IN PARTNERSHIP WITH:
CNRS

Université Denis Diderot (Paris 7)

Activity Report 2014

Project-Team GANG

Networks, Graphs and Algorithms

IN COLLABORATION WITH: Laboratoire d'Informatique Algorithmique Fondamentale et Appliquée (LIAFA)

RESEARCH CENTER
Paris - Rocquencourt

THEME
Networks and Telecommunications

Table of contents

Project-Team GANG

Keywords: Graph Theory, Dynamic Networks, Fault Tolerance, Peer-to-peer, Distributed Algorithms

Creation of the Project-Team: 2007 July 01.

1. Members

Research Scientists

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

Faculty Members

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

Engineer

François Durand [Inria]

PhD Students

The-Dang Huynh [Bell Labs (Alcatel)]
Leonardo Linguaglossa [Inria]

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 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 underlied 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 other's 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. Application Domains

Application domains include evaluating Internet performances, the design of new peer-to-peer applications, enabling large scale ad hoc networks and mapping the web.

- The application of measuring and modeling Internet metrics such as latencies and bandwidth is to provide tools for optimizing Internet applications. This concerns especially large scale applications such as web site mirroring and peer-to-peer applications.
- Peer-to-peer protocols are based on a all equal paradigm that allows to design highly reliable and scalable applications. Besides the file sharing application, peer-to-peer solutions could take over in web content dissemination resistant to high demand bursts or in mobility management. Envisioned peer-to-peer applications include video on demand, streaming, exchange of classified ads,...
- Wifi networks have entered our every day life. However, enabling them at large scale is still a challenge. Algorithmic breakthrough in large ad hoc networks would allow to use them in fast and economic deployment of new radio communication systems.
- The main application of the web graph structure consists in ranking pages. Enabling site level indexing and ranking is a possible application of such studies.

5. New Results

5.1. Highlights of the Year

Pierre Fraigniaud has received the Prize for Innovation in Distributed Computing 2014.

5.2. Graph and Combinatorial Algorithms

5.2.1. *Collision-Free Network Exploration*

In the collision-free exploration model considered in [16], a set of mobile agents is placed at different nodes of a n-node network. The agents synchronously move along the network edges in a collision-free way, i.e., in no round may two agents occupy the same node. In each round, an agent may choose to stay at its currently occupied node or to move to one of its neighbors. An agent has no knowledge of the number and initial positions of other agents. We are looking for the shortest possible time required to complete the collision-free *network exploration*, i.e., to reach a configuration in which each agent is guaranteed to have visited all network nodes and has returned to its starting location.

In this work, we first considered the scenario when each mobile agent knows the map of the network, as well as its own initial position. We established a connection between the number of rounds required for collision-free exploration and the degree of the minimum-degree spanning tree of the graph. We provided tight (up to a constant factor) lower and upper bounds on the collision-free exploration time in general graphs, and the exact value of this parameter for trees. For our second scenario, in which the network is unknown to the agents, we proposed collision-free exploration strategies running in $O(n^2)$ rounds for tree networks and in $O(n^5 \log n)$ rounds for general networks.

5.2.2. Properties of Graph Search Procedures

In [4], we study the last vertex discovered by a graph search such as BFS or DFS. End-vertices of a given graph search may have some nice properties (as for example it is well known that the last vertex of Lexicographic Breadth First Search (LBFS) in a chordal graph is simplicial). Therefore it is interesting to consider if these vertices can be recognized in polynomial time or not. A graph search is a mechanism for systematically visiting the vertices of a graph. At each step of a graph search, the key point is the choice of the next vertex to be explored. Graph searches only differ by this selection mechanism during which a tie-break rule is used. In this paper we study how the choice of the tie-break rule can determine the complexity of the end-vertex problem for BFS or DFS. In particular we prove a counter-intuitive NP-completeness result for Breadth First Search, answering a question of D.G. Corneil, E. Köhler and J-M Lanlignel.

5.2.3. Matchings in Hypergraphs

A rainbow matching for (not necessarily distinct) sets $F_1, ...F_k$ of hypergraph edges is a matching consisting of k edges, one from each F_i. The aim of [3] is twofold—to put order in the multitude of conjectures that relate to this concept (some first presented here), and to prove partial results on one of the central conjectures settled by Ryser, Brualdi and Stein.

5.2.4. Common Intervals and Application to Genome Comparison

In [6], we show how to identify generalized common and conserved nested intervals. This is a bio-informatics papers, explaining how to compute more relaxed variants of common or of conserved intervals of two permutations, which has applications in genome comparison. It also presents some properties of the family of intervals, useful for storing them.

5.2.5. Graph Decomposition

In [10], we present a general framework for computing a large family of graph decomposition, the H-join. It generalizes some well know tools like modular decomposition or split decomposition. The paper explains how to compute it in polynomial time. A new canonical decomposition for sesquiprime graphs is also presented.

5.2.6. Combinatorial Optimization

Normal cone and subdifferential have been generalized through various continuous functions; in [8], we focus on a non separable Q-subdifferential version. Necessary and sufficient optimality conditions for unconstrained nonconvex problems are revisited accordingly. For inequality constrained problems, Q-subdifferential and the lagrangian multipliers, enhanced as continuous functions instead of scalars, allow us to derive new necessary and sufficient optimality conditions. In the same way, the Legendre-Fenchel conjugate is generalized into Q-conjugate and global optimality conditions are derived by Q-conjugate as well, leading to a tighter inequality.

5.3. Distributed Computing

5.3.1. Rendezvous

5.3.1.1. Rendezvous of Anonymous Agents in Trees

In [5], we study the so-called *rendezvous problem* in the mobile agent setting in graph environments. In the studied model, two identical (anonymous) mobile agents start from arbitrary nodes of an unknown tree and have to meet at some node. Agents move in synchronous rounds: in each round an agent can either stay at the current node or move to one of its neighbors. We consider deterministic algorithms for this rendezvous task. The main result of our research is a tight trade-off between the optimal time of completing rendezvous and the size of memory of the agents. For agents with k memory bits, we show that optimal rendezvous time is $\Theta(n + n^2/k)$ in n-node trees. More precisely, if $k \geq c \log n$, for some constant c, we design agents accomplishing rendezvous in arbitrary trees of size n (unknown to the agents) in time $O(n + n^2/k)$, starting with arbitrary delay. We also show that no pair of agents can accomplish rendezvous in time $o(n + n^2/k)$, even in the class of lines of known length and even with simultaneous start. Finally, we prove that at least logarithmic memory is necessary for rendezvous, even for agents starting simultaneously in a n-node line.

5.3.1.2. Rendezvous of Distance-Aware Mobile Agents in Unknown Graphs

In [17], we study the problem of rendezvous of two mobile agents starting at distinct locations in an unknown graph. The agents have distinct labels and walk in synchronous steps. However, the graph is unlabeled and the agents have no means of marking the nodes of the graph and cannot communicate with or see each other until they meet at a node. When the graph is very large, we would like the time to rendezvous to be independent of the graph size and to depend only on the initial distance between the agents and some local parameters such as the degree of the vertices, and the size of the agent's label. It is well known that even for simple graphs of degree Δ, the rendezvous time can be exponential in Δ in the worst case. In this study, we introduce a new version of the rendezvous problem where the agents are equipped with a device that measures its distance to the other agent after every step. We show that these *distance-aware* agents are able to rendezvous in any unknown graph, in time polynomial in all the local parameters such the degree of the nodes, the initial distance D and the size of the smaller of the two agent labels $l = \min(l_1, l_2)$. Our algorithm has a time complexity of $O(\Delta(D + \log l))$ and we show an almost matching lower bound of $\Omega(\Delta(D + \log l / \log \Delta))$ on the time complexity of any rendezvous algorithm in our scenario. Further, this lower bound extends existing lower bounds for the general rendezvous problem without distance awareness.

5.3.1.3. Rendezvous of Heterogeneous Mobile Agents in Edge-Weighted Networks

In [22], we study the deterministic rendezvous problem in which a pair of heterogeneous agents, differing in the time required to traverse particular edges of the graph, need to meet on an edge or node of the graph. Each of the agents knows the complete topology of the undirected graph and the initial positions of both of the agents. The agent also knows its own traversal times for all of the edges of the graph, but is unaware of the corresponding traversal times for the other agent. In this scenario, we study the time required by the agents to meet, compared to the time T_{OPT} in the offline scenario in which the agents have complete knowledge of each others capabilities. When no additional assumptions are made, we show that rendezvous can be achieved after time $O(nT_{\mathrm{OPT}})$ in a n-node graph, and that this time is essentially the best possible in some cases. However, the rendezvous time can be reduced to $\Theta(T_{\mathrm{OPT}})$ when the agents are allowed to exchange $\Theta(n)$ bits of information at the start of the rendezvous process. We then show that under some natural assumption about the traversal times of edges, the hardness of the heterogeneous rendezvous problem can be substantially decreased, both in terms of time required for rendezvous without communication, and the communication complexity of achieving rendezvous in time $\Theta(T_{\mathrm{OPT}})$.

5.3.1.4. Rendezvous with Different Speeds

In [32] we introduce the study of the rendezvous problem in the context of agents having different speeds, and present tight and almost tight bounds for this problem, restricted to a ring topology.

5.3.2. Fair Synchronization

A non-blocking implementation of a concurrent object is an implementation that does not prevent concurrent accesses to the internal representation of the object, while guaranteeing the deadlock-freedom progress condition without using locks. Considering a failure free context, G. Taubenfeld has introduced (DISC 2013) a simple modular approach, captured under a new problem called the *fair synchronization* problem, to transform a non-blocking implementation into a starvation-free implementation satisfying a strong fairness requirement.

This approach is illustrated in [19] with the implementation of a concurrent stack. The spirit of the paper is mainly pedagogical. Its aim is not to introduce new concepts or algorithms, but to show that a powerful, simple, and modular transformation can provide concurrent objects with strong fairness properties.

In [20], we extend this approach in several directions. It first generalizes the fair synchronization problem to read/write asynchronous systems where any number of processes may crash. Then, it introduces a new failure detector and uses it to solve the fair synchronization problem when processes may crash. This failure detector, denoted QP (Quasi Perfect), is very close to, but strictly weaker than, the perfect failure detector. Last but not least, the paper shows that the proposed failure detector QP is optimal in the sense that the information on failures it provides to the processes can be extracted from any algorithm solving the fair synchronization problem in the presence of any number of process crash failures.

5.3.3. Wait Free with Advice

In [7], we motivate and propose a new way of thinking about failure detectors which allows us to define, quite surprisingly, what it means to solve a distributed task *wait-free using a failure detector*. In our model, the system is composed of *computation* processes that obtain inputs and are supposed to produce outputs and *synchronization* processes that are subject to failures and can query a failure detector.

Under the condition that *correct* synchronization processes take sufficiently many steps, they provide the computation processes with enough *advice* to solve the given task wait-free: every computation process outputs in a finite number of its own steps, regardless of the behavior of other computation processes.

Every task can thus be characterized by the *weakest* failure detector that allows for solving it, and we show that every such failure detector captures a form of set agreement. We then obtain a complete classification of tasks, including ones that evaded comprehensible characterization so far, such as renaming or weak symmetry breaking.

5.3.4. Adaptive Register Allocation

In [18], we give an adaptive algorithm in which processes use multi-writer multi-reader registers to acquire exclusive write access to their own single-writer, multi-reader registers. It is the first such algorithm that uses a number of registers linear in the number of participating processes. Previous adaptive algorithms require at least $\Theta(n^{3/2})$ registers.

5.3.5. Leader Election

Considering the case of homonyms processes (some processes may share the same identifier) on a ring [21], we give a necessary and sufficient condition on the number of identifiers to enable leader election. We prove that if l is the number of identifiers then message-terminating election is possible if and only if l is greater than the greatest proper divisor of the ring size even if the processes do not know the ring size. If the ring size is known, we propose a process-terminating algorithm exchanging $O(n\log(n))$ messages that is optimal.

5.3.6. Concurrency and Fault-tolerance

In [15], we study the connections between self-stabilization and proof-labeling schemes. It follows from the definition of *silent* self-stabilization, and from the definition of *proof-labeling* scheme, that if there exists a silent self-stabilizing algorithm using ℓ-bit registers for solving a task T, then there exists a proof-labeling scheme for T using registers of at most ℓ bits. The first result in this paper is the converse to this statement. We show that if there exists a proof-labeling scheme for a task T, using ℓ-bit registers, then there exists a silent self-stabilizing algorithm using registers of at most $O(\ell+\log n)$ bits for solving T, where n is the number of processes in the system. Therefore, as far as memory space is concerned, the design of silent self-stabilizing algorithms essentially boils down to the design of compact proof-labeling schemes. The second result in this paper addresses time complexity. We show that, for every task T with k-bits output size in n-node networks, there exists a silent self-stabilizing algorithm solving T in $O(n)$ rounds, using registers of $O(n^2+kn)$ bits. Therefore, as far as running time is concerned, *every* task has a silent self-stabilizing algorithm converging in a linear number of rounds.

In [27], we study the connections between, on the one hand, asynchrony and concurrency, and, on the other hand, the quality of the expected solution of a distributed algorithm. The state machine approach is a well-known technique for building distributed services requiring high performance and high availability, by replicating servers, and by coordinating client interactions with server replicas using consensus. Indulgent consensus algorithms exist for realistic eventually partially synchronous models, that never violate safety and guarantee liveness once the system becomes synchronous. Unavoidably, these algorithms may never terminate, even when no processor crashes, if the system never becomes synchronous. We propose a mechanism similar to state machine replication, called *RC-simulation*, that can always make progress, even if the system is never synchronous. Using RC-simulation, the quality of the service will adjust to the current level of asynchrony of the network — degrading when the system is very asynchronous, and improving when the system becomes more synchronous. RC-simulation generalizes the state machine approach in the following sense: when the system is asynchronous, the system behaves as if $k+1$ threads were running concurrently, where k is a function of the asynchrony. In order to illustrate how the RC-simulation can be used, we describe a long-lived renaming implementation. By reducing the concurrency down to the asynchrony of the system, RC-simulation enables to obtain renaming quality that adapts linearly to the asynchrony.

5.3.7. Quantum Computing

In [1], we provide illustrative examples of distributed computing problems for which it is possible to design tight lower bounds for *quantum* algorithms without having to manipulate concepts from quantum mechanics, at all. As a case study, we address the following class of 2-player problems. Alice (resp., Bob) receives a boolean x (resp., y) as input, and must return a boolean a (resp., b) as output. A *game* between Alice and Bob is defined by a pair (δ, f) of boolean functions. The objective of Alice and Bob playing game (δ, f) is, for every pair (x, y) of inputs, to output values a and b, respectively, satisfying $\delta(a, b) = f(x, y)$, in *absence of any communication* between the two players, but in *presence of shared resources*. The ability of the two players to solve the game then depends on the type of resources they share. It is known that, for the so-called CHSH game, i.e., for the game $a \oplus b = x \wedge y$, the ability for the players to use entangled quantum bits (qubits) helps. We show that, apart from the CHSH game, quantum correlations do not help, in the sense that, for every game not equivalent to the CHSH game, there exists a classical protocol (using shared randomness) whose probability of success is at least as large as the one of any protocol using quantum resources. This result holds for both worst case and average case analysis. It is achieved by considering a model stronger than quantum correlations, the *non-signaling model*, which subsumes quantum mechanics, but is far easier to handle.

5.3.8. Distributed Decision and Verification

5.3.8.1. Randomization

In [12], we study the power of randomization in the context of locality by analyzing the ability to “boost” the success probability of deciding a distributed language. The main outcome of this analysis is that the distributed computing setting contrasts significantly with the sequential one as far as randomization is concerned. Indeed, we prove that in some cases, the ability to increase the success probability for deciding distributed languages is rather limited.

5.3.8.2. Model Variants

In a series of papers [14], [28], we analyze distributed decision in the context of various models for distributed computing.

In [28], we carry on the effort to bridging runtime verification with distributed computability, studying necessary conditions for monitoring failure prone asynchronous distributed systems. It has been recently proved that there are correctness properties that require a large number of opinions to be monitored, an opinion being of the form true, false, perhaps, probably true, probably no, etc. The main outcome of this paper is to show that this large number of opinions is not an artifact induced by the existence of artificial constructions. Instead, monitoring an important class of properties, requiring processes to produce at most k different values does require such a large number of opinions. Specifically, our main result is a proof that it is impossible to monitor k-set-agreement in an n-process system with fewer than $\min\{2k, n\} + 1$ opinions. We also provide an algorithm to monitor k-set-agreement with $\min\{2k, n\} + 1$ opinions, showing that the lower bound is tight.

Finally, in [14], we tackle *local distributed testing* of graph properties. This framework is well suited to contexts in which data dispersed among the nodes of a network can be collected by some central authority (like in, e.g., sensor networks). In local distributed testing, each node can provide the central authority with just a few information about what it perceives from its neighboring environment, and, based on the collected information, the central authority is aiming at deciding whether or not the network satisfies some property. We analyze in depth the prominent example of checking *cycle-freeness*, and establish tight bounds on the amount of information to be transferred by each node to the central authority for deciding cycle-freeness. In particular, we show that distributively testing cycle-freeness requires at least $\lceil \log d \rceil - 1$ bits of information per node in graphs with maximum degree d, even for connected graphs. Our proof is based on a novel version of the seminal result by Naor and Stockmeyer (1995) enabling to reduce the study of certain kinds of algorithms to order-invariant algorithms, and on an appropriate use of the known fact that every free group can be linearly ordered.

5.3.9. Voting Systems

In [44], [38], we consider a general framework for voting systems with arbitrary types of ballots such as orders of preference, grades, etc. We investigate their manipulability: in what states of the population may a coalition of electors, by casting an insincere ballot, secure a result that is better from their point of view?

We show that, for a large class of voting systems, a simple modification allows to reduce manipulability. This modification is *Condorcification*: when there is a Condorcet winner, designate her; otherwise, use the original rule.

When electors are independent, for any non-ordinal voting system (i.e. requiring information that is not included in the orders of preferences, for example grades), we prove that there exists an ordinal voting system whose manipulability rate is at most as high and which meets some other desirable properties. Furthermore, this result is also true when voters are not independent but the culture is *decomposable*, a weaker condition that we define.

Combining both results, we conclude that when searching for a voting system whose manipulability is minimal (in a large class of systems), one can restrict to voting systems that are ordinal and meet the Condorcet criterion.

In [35], we examine the geometrical properties of the space of expected utilities over a finite set of options, which is commonly used to model the preferences of an agent. We focus on the case where options are assumed to be symmetrical a priori, which is a classical neutrality assumption when studying voting systems. Specifically, we prove that the only Riemannian metric that respects the geometrical properties and the natural symmetries of the utility space is the round metric. Whereas Impartial Culture is widely used in Social Choice literature but limited to ordinal preference, our theoretical result allows to extend it canonically to cardinal preferences.

In [25], we study the manipulability of voting systems in a real-life experiment: electing the best paper in the conference Algotel 2012. Based on real ballots, we provide a quantitative study of the manipulability, as a function of the voting system used. We show that, even in a situation where all voting systems give the same winner by sincere voting, choosing the voting system is critical, because it has a huge impact on manipulability. In particular, one voting system fare way be better than the others: Instant-Runoff Voting.

5.4. Network Algorithms and Analysis

5.4.1. Bounds on the Cover Time in the Rotor-Router Model

In [23] and [33], we consider the *rotor-router mechanism*, which provides a deterministic alternative to the random walk in undirected graphs. In this model, a set of k identical walkers is deployed in parallel, starting from a chosen subset of nodes, and moving around the graph in synchronous steps. During the process, each node maintains a cyclic ordering of its outgoing arcs, and successively propagates walkers which visit it along its outgoing arcs in round-robin fashion, according to the fixed ordering. 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 starting locations of the walks. In the case of $k = 1$, Yanovski et al. (2003) and Bampas et al. (2009) showed that a single walk achieves a cover time of exactly $\Theta(mD)$ for any n-node graph with m edges and diameter D, and that the walker explores increasingly large Eulerian subgraphs before eventually stabilizes to a traversal of an Eulerian circuit on the set of all directed edges of the graph.

In [23], we provide tight bounds on the cover time of k parallel rotor walks in a graph. We show that this cover time is at most $\Theta(mD/\log k)$ and at least $\Theta(mD/k)$ for any graph, which corresponds to a speedup of between $\Theta(\log k)$ and $\Theta(k)$ with respect to the cover time of a single walk. Both of these extremal values of speedup are achieved for some graph classes. Our results hold for up to a polynomially large number of walks, $k = O(poly(n))$.

In [33], we perform a case study of cover time of the rotor-router, showing how the cover time depends on k for many important graph classes. We determine the precise asymptotic value of the rotor-router cover time for all values of k for degree-restricted expanders, random graphs, and constant-dimensional tori. For hypercubes, we also resolve the question precisely, except for values of k much larger than n. Our results can be compared to those obtained by Elsässer and Sauerwald (2009) in an analogous study of the cover time of k independent parallel random walks in a graph; for the rotor-router, we obtain tight bounds in a slightly broader spectrum of cases. Our proofs take advantage of a relation which we develop, linking the cover time of the rotor-router to the mixing time of the random walk and the local divergence of a discrete diffusion process on the considered graph.

5.4.2. Web Ranking and Aliveness

In [29] and [30], we investigate how to efficiently retrieve large portions of alive pages from an old crawl using orderings we called LiveRanks. Our work establishes the possibility of efficiently recovering a significant portion of the alive pages of an old snapshot and advocates for the use of an adaptive sample-based PageRank for obtaining an efficient LiveRank. Additionally, application field is not limited to Web graphs. It can be straightforwardly adapted to any online data with similar linkage enabling crawling, like P2P networks or online social networks.

5.4.3. Wireless Positioning

In [31], we consider how to construct a low-cost and efficient positioning system. We have proposed a new method called Two-Step Movement (2SM) to estimate the position of Mobile Terminal (MT). By exploiting useful information given by the position change of the device or user movement, this method can minimize the number of Reference Points (RP) required (*i.e.,* only one) in a localization system or navigation service and reduce system implementation cost. Analytical result shows that the user position can be derived, under noisy environment, with an estimation error about 10% of the distance between the RP and MT, or even less.

5.4.4. Content Centric Networking

Today's Internet usage is mostly centered around location-independent services. Because the Internet architecture is host-centric, content or service requests still have to be translated into locations, or the IP address of their hosts. This translation is realized through different technologies, e.g. DNS and HTTP redirection, which are currently implemented at the Application Layer. (ICN) proposes to evolve the current Internet infrastructure by extending the networking layer with name-based primitives.

In [45], we target the design and implementation of a content router, which is a network entity that implements *name-based forwarding*, or it can forward packets based on the content name they are addressed to. This work makes three major contributions. First, we propose an algorithm for name-based longest prefix match whose main novelty is the *prefix Bloom filter*, a Bloom filter variant that exploits the hierarchical nature of content prefixes. Second, a content router design that is compatible with both today's networking protocols and with widely used network equipments. Third, two innovative features that increase the scalability of a content router both in term of forwarding-information-base size and forwarding speed.

In the demonstration [34] held in the ICN conference, we demonstrate a high speed Information-Centric Network in a mobile backhaul setting. In particular, we emulate an information aware data plane and we highlight the significant benefits it provides in terms of both user experience and network provider cost in the backhaul setting. Our setup consists of high-speed ICN devices employed in a down-scaled realistic representation of a mobile backhaul topology, fed with traffic workloads characterized from Orange's mobile network. We compare numerical results activating and de-activating the ICN feature at run-time, showing the main differences between the two approaches. All the devices are implemented in a real high-speed multi-core equipment, and they are connected by means of internal port connections. Traffic is injected using a Traffic Generator which is implemented in the same architecture.

5.4.5. Information Dissemination

5.4.5.1. Dissemination with Noise or Limited Memory

In [26], we introduce the study of basic distributed computing problems in the context of noise in communication. We establish tight and almost tight bounds for the rumor spreading problem as well as for the majority-consensus problem.

In [11], we theoretically study a general model of information sharing within animal groups. We take an algorithmic perspective to identify efficient communication schemes that are, nevertheless, economic in terms of communication, memory and individual internal computation. We present a simple and natural algorithm in which each agent compresses all information it has gathered into a single parameter that represents its confidence in its behavior. Confidence is communicated between agents by means of active signaling. We motivate this model by novel and existing empirical evidences for confidence sharing in animal groups. We rigorously show that this algorithm competes extremely well with the best possible algorithm that operates without any computational constraints. We also show that this algorithm is minimal, in the sense that further reduction in communication may significantly reduce performances. Our proofs rely on the Cramér-Rao bound and on our definition of a Fisher Channel Capacity. We use these concepts to quantify information flows within the group which are then used to obtain lower bounds on collective performance.

5.4.5.2. Gossip and Rumor Spreading with Flooding

In [2], we address the flooding problem in dynamic graphs, where flooding is the basic mechanism in which every node becoming aware of an information at step t forwards this information to all its neighbors at all forthcoming steps $t' > t$. In particular, we show that a technique developed in a previous paper, for analyzing flooding in a Markovian sequence of Erdös-Rényi graphs, is robust enough to be used also in different contexts. We establish this by analyzing flooding in a sequence of graphs drawn independently at random according to a model of random graphs with given expected degree sequence. In the prominent case of power-law degree distributions, we prove that flooding takes almost surely $O(\log n)$ steps even if, almost surely, none of the graphs in the sequence is connected. In the general case of graphs with an arbitrary degree sequence, we prove several upper bounds on the flooding time, which depend on specific properties of the degree sequence.

5.4.6. Small-world Networks

In [9], we study decentralized routing in small-world networks that combine a wide variation in node degrees with a notion of spatial embedding. Specifically, we consider a variant of J. Kleinberg's grid-based small-world model in which (1) the number of long-range edges of each node is not fixed, but is drawn from a power-law probability distribution with exponent parameter $\alpha \geq 0$ and constant mean, and (2) the long-range edges are considered to be bidirectional for the purposes of routing. This model is motivated by empirical observations indicating that several real networks have degrees that follow a power-law distribution. The measured power-law exponent α for these networks is often in the range between 2 and 3. For the small-world model we consider, we show that when $2 < \alpha < 3$ the standard greedy routing algorithm, in which a node forwards the message to its neighbor that is closest to the target in the grid, finishes in an expected number of $O(\log^{\alpha-1} n \cdot \log\log n)$ steps, for any source–target pair. This is asymptotically smaller than the $O(\log^2 n)$ steps needed in Kleinberg's original model with the same average degree, and approaches $O(\log n)$ as α approaches 2. Further, we show that when $0 \leq \alpha < 2$ or $\alpha \geq 3$ the expected number of steps is $O(\log^2 n)$, while for $\alpha = 2$ it is $O(\log^{4/3} n)$. We complement these results with lower bounds that match the upper bounds within at most a $\log\log n$ factor.

5.4.7. Voting Systems and Path Selection in Networks

In [24], we apply our theoretical and experimental results on voting systems to a network use case: choosing a path in a network. In our model, nodes have an economical reward or cost for each possible path and they vote to elect the path. We show that the choice of the voting system has an important impact on the manipulability and the economical efficiency of this system. From both points of view, Instant-Runoff Voting gives the best results.

6. Bilateral Contracts and Grants with Industry

6.1. Alcatel-Lucent Bell Labs

Participants: François Durand, The-Dang Huynh, Leonardo Linguaglossa, Laurent Viennot.

Gang has a strong collaboration with Alcatel-Lucent. We notably collaborate with Fabien Mathieu and Diego Perino who are former members of Gang that have joined Alcatel-Lucent. A Cifre grant allows to fund the PhD thesis of The-Dang Huynh to study ranking techniques and their application to social networks. An ADR (joint research action) is dedicated to content centric networks and forwarding information verification. The PhD thesis of Leonardo Linguaglossa is funded by this contract. We also collaborate with Ludovic Noirie on voting systems.

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

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. 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 Ph. D.

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.

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

Participants: François Durand, The-Dang Huynh, Leonardo Linguaglossa, Laurent Viennot.

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.

7.2. International Initiatives

7.2.1. Inria International Partners

7.2.1.1. Informal International Partners

- Carole Delporte and Hugues Fauconnier collaborate with Sam Toueg (Univ. of Toronto) and Rachid Guerraoui (EPFL) on distributed computing and synchronization.
- Carole Delporte, Hugues Fauconnier and Pierre Fraigniaud collaborate on distributed computing with Eli Gafni (UCLA) and Sergio Rajsbaum (Univ. of Mexico).
- Pierre Fraigniaud collaborates with Zvi Lotker (Ben-Gurion Univ.) on social networks.
- Amos Korman collaborates with Ofer Feinerman (Weizmann Institute) on the application of distributed algorithm analysis to ant behaviors.

7.3. International Research Visitors

7.3.1. Visits of International Scientists

- Eli Gafni, UCLA, June - July 2014

- Sergio Rajsbaum, Univ. of Mexico, June - July 2014
- Zvi Lotker, Ben-Gurion Univ., September 2014 - July 2015 (Junior chair of the FSMP)

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific events organisation

8.1.1.1. General chair, scientific chair

Amos Korman has co-chaired and co-organized the BDA 2014 worksop http://www.cs.cmu.edu/~saketn/BDA2014/.This 2 day workshop brought together leading researchers in biology and distributed computing (from institutions such as MIT, Harvard, Stanford, and Weizmann) aiming to create a common platform for collaborations.

8.1.2. Scientific events selection

8.1.2.1. Chair of conference program committee

Pierre Fraigniaud has chaired the C track of the 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014).

8.1.2.2. Member of the conference program committee

Michel Habib was member of the program committee of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS 2014).

8.1.2.3. Invited talks

Pierre Fraigniaud was keynote speaker at the 16th Interntional Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2014), Paderborn, Germany, Sep 28 - Oct 1, 2014.

Adrian Kosowski made a tutorial at the 5th Polish Combinatorial Conference, Bedlewo, September 2014.

Adrian Kosowski was invited speaker at Algotel 2014 - 16èmes Rencontres Francophones pour les Aspects Algorithmiques des Télécommunications, Ile de Ré, June 2014.

Laurent Viennot was invited speaker at the SMBE Satellite Meeting on Reticulated Microbial Evolution, April 27-30, 2014, Kiel, Germany.

Michel Habib made a tutorial at Ecole Rescom, France, May 2014.

8.1.3. Journal

8.1.3.1. Member of the editorial board

Pierre Fraigniaud is member of the editorial boards of Distributed Computing (DC), Theory of Computing Systems (TOCS), Fundamenta Informaticae (FI) and Journal of Interconnection Networks (JOIN).

8.2. Teaching - Supervision - Juries

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

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

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

Master: Adrian Kosowski, Communication and Routing, 4.5h, M1, ENSEIRB-MATMECA

Master: Adrian Kosowski, Randomization in Computer Science: Games, Networks, Epidemic and Evolutionary Algorithms, 24h, M1, École Polytechnique

Master: Adrian Kosowski, Distributed Computing with Mobile Agents: Exploration, Rendezvous, and related problems, 12h, M2, IX Summer School on Discrete Mathematics in South America, Valparaiso, Chile

Licence : Fabien de Montgolfier, Introduction à la programmation, 26h, L1, Univ Paris Diderot

Licence : Fabien de Montgolfier, Programmation avancée (bio-informatique), 26h, L3, Univ. Paris Diderot

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

Licence : Fabien de Montgolfier, Systèmes et Réseaux, 52h, L3, Ecole d'Ingénieurs Denis Diderot

Master : Laurent Viennot, Système, réseau et Internet, 15h, M1, Univ. Paris Diderot

License : 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, 18h, M2 MPRI, Université Paris Diderot, France

8.2.2. Supervision

PhD: Jérémie Dusart, Parcours de graphes, applications aux graphes de cocomparabilité, Université Paris Diderot, defended June 2014, supervised by Michel Habib.

PhD: Antoine Mamcarz, Décompositions de trigraphes et parcours de graphes, Université Paris Diderot, defended June 30, 2014, supervised by Michel Habib.

PhD: Dominik Pajak, Algorithms for Deterministic Parallel Graph Exploration, Université de Bordeaux, defended June 13, 2014, supervised by Adrian Kosowski and Ralf Klasing.

PhD: The-Dang Huynh, Extensions de PageRank et Applications aux Réseaux Sociaux, Université Pierre et Marie Curie, since 2012, supervised by Fabien Mathieu, Dohy Hong and Laurent Viennot.

PhD: Leonardo Linguaglossa, Design of algorithms and protocols to support ICN functionalities in high speed routers, Université Paris Diderot, since 2013, supervised by Fabien Mathieu, Diego Perino and Laurent Viennot.

8.2.3. Juries

PhD : Hugues Fauconnier was member of the jury of Peva Blanchard, Synchronization and Fault-tolerance in Distributed Algorithms, Univ. Paris XI, September 24th

PhD : Laurent Viennot was reviewer of the thesis of Pierre-Alain Jachiet, Étude de l'évolution combinatoire des gènes par l'analyse de réseaux de similarité de séquence, Université Pierre et Marie Curie, July 2014, supervised by Eric Bapteste and Philippe Lopez

PhD : Laurent Viennot was reviewer of the thesis of Arnaud Jégou, Harnessing the power of implicit and explicit social networks through decentralization, Université de Rennes 1, September 2014, supervised by Anne-Marie Kermarrec and Davide Frey

PhD : Laurent Viennot was reviewer of the thesis of Pierre Halftermeyer, Connexité dans les réseaux et schémas d'étiquetage compact d'urgence, Université de Bordeaux, September 2014, supervised by Bruno Courcelle and Cyril Gavoille

PhD : Laurent Viennot was reviewer of the thesis of Aurélien Lancin, Étude de réseaux complexes et de leurs propriétés pour l'optimisation de modèles de routage, Université de Nice – Sophia Antipolis, December 2014, supervised by David Coudert

PhD : Fabien de Montgolfier was member of the jury of Pierre Clairet, Approche algorithmique pour l'amélioration des performances du système de détection d'intrusions PIGA, Université d'Orléans, defended on June 24th, 2014, supervised by Pascal Berthomé.

PhD : Fabien de Montgolfier was member of the jury of Antoine Mamcarz, Décompositions de trigraphes et parcours de graphes, Université Paris Diderot, defended on June 30th, 2014, supervised by Michel Habib.

8.3. Popularization

Laurent Viennot has written an article on the history of telecommunication networks with a special focus on the phone network and on Internet [13].

Laurent Viennot has animated a weekly workshop on computer science unplugged in the Pouchet primary school in Paris at CM1 level (15h during 2014). The topics covered included Nim game, Euler paths in graphs, and error correcting codes.

9. Bibliography

Publications of the year

Articles in International Peer-Reviewed Journals

[1] H. ARFAOUI, P. FRAIGNIAUD. *What can be computed without communications?*, in "ACM SIGACT News", 2014, vol. 45, n^{o} 3, pp. 82-104 [*DOI :* 10.1145/2670418.2670440], https://hal.inria.fr/hal-01102110

[2] H. BAUMANN, P. CRESCENZI, P. FRAIGNIAUD. *Flooding in dynamic graphs with arbitrary degree sequence*, in "Journal of Parallel and Distributed Computing", 2014, vol. 74, n^{o} 5, pp. 2433-2437 [*DOI :* 10.1016/J.JPDC.2014.01.007], https://hal.inria.fr/hal-01102106

[3] P. CHARBIT, R. AHARONI, D. HOWARD. *On a Generalization of the Ryser-Brualdi-Stein Conjecture*, in "Journal of Graph Theory", January 2015, 14 p. [*DOI :* 10.1002/JGT.21796], https://hal.inria.fr/hal-01101508

[4] P. CHARBIT, M. HABIB, A. MAMCARZ. *Influence of the tie-break rule on the end-vertex problem*, in "Discrete Mathematics and Theoretical Computer Science", 2014, 16 p. , https://hal.inria.fr/hal-01101514

[5] J. CZYZOWICZ, A. KOSOWSKI, A. PELC. *Time versus space trade-offs for rendezvous in trees*, in "Distributed Computing", April 2014, vol. 27, n^{o} 2, pp. 95-109 [*DOI :* 10.1007/S00446-013-0201-4], https://hal.inria.fr/hal-00646912

[6] F. DE MONTGOLFIER, M. RAFFINOT, I. RUSU. *Easy identification of generalized common and conserved nested intervals*, in "Journal of Computational Biology", July 2014, vol. 21, n^{o} 7, pp. 520-533 [*DOI :* 10.1089/CMB.2013.0146], https://hal.archives-ouvertes.fr/hal-00921762

[7] C. DELPORTE-GALLET, E. GAFNI, H. FAUCONNIER, P. KOUZNETSOV. *Wait-freedom with advice*, in "Distributed Computing", 2014, vol. on line, 20 p. , https://hal.inria.fr/hal-01100779

[8] D. FORTIN, I. TSEVEENDORJ. *Q-subdifferential and Q-conjugate for global optimality*, in "Computational Mathematics and Mathematical Physics / Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki", February 2014, vol. 54, n^{o} 2, pp. 265-274, https://hal.inria.fr/hal-00957373

[9] P. FRAIGNIAUD, G. GIAKKOUPIS. *Greedy routing in small-world networks with power-law degrees*, in "Distributed Computing", 2014, vol. 27, n^{o} 4, pp. 231 - 253 [*DOI :* 10.1007/S00446-014-0210-Y], https://hal.inria.fr/hal-01097141

[10] M. HABIB, A. MAMCARZ, F. DE MONTGOLFIER. *Computing H-Joins with Application to 2-Modular Decomposition*, in "Algorithmica", 2014, vol. 70, n^{o} 2, pp. 245-266 [*DOI :* 10.1007/S00453-013-9820-1], https://hal.archives-ouvertes.fr/hal-00921775

[11] A. KORMAN, G. EFRAT, O. FEINERMAN. *Confidence Sharing: An Economic Strategy for Efficient Information Flows in Animal Groups*, in "PLoS Computational Biology", October 2014, 10 p. [*DOI :* 10.1371/JOURNAL.PCBI.1003862], https://hal.inria.fr/hal-01093295

[12] P. MERAV, D. PELEG, A. KORMAN, P. FRAIGNIAUD, G. MIKA. *Randomized distributed decision*, in "Distributed Computing", December 2014, 15 p. [*DOI :* 10.1007/S00446-014-0211-X], https://hal.inria.fr/hal-01093255

Articles in National Peer-Reviewed Journals

[13] L. VIENNOT. *Une brève histoire des réseaux de télécommunications*, in "Interstices", June 2014, 1 p. , https://hal.inria.fr/hal-01104358

Invited Conferences

[14] H. ARFAOUI, P. FRAIGNIAUD, D. ILCINKAS, F. MATHIEU. *Distributedly Testing Cycle-Freeness*, in "Proceedings of the 40th International Workshop on Graph-Theoretic Concepts in Computer Science", Nouan-le-Fuzelier, France, LNCS, June 2014, vol. 8747, pp. 15 - 28 [*DOI :* 10.1007/978-3-319-12340-0_2], https://hal.archives-ouvertes.fr/hal-01084297

International Conferences with Proceedings

[15] L. BLIN, P. FRAIGNIAUD, B. PATT-SHAMIR. *On Proof-Labeling Schemes versus Silent Self-stabilizing Algorithms*, in "SSS - 16th International Symposium on Stabilization, Safety, and Security of Distributed Systems", Paderborn, Germany, P. FELBER, V. GARG (editors), LNCS - Lecture Notes in Computer Science, Springer , September 2014, vol. 8756, pp. 18-32 [*DOI :* 10.1007/978-3-319-11764-5_2], https://hal.inria.fr/hal-01102123

[16] J. CZYZOWICZ, D. DERENIOWSKI, L. GASIENIEC, R. KLASING, A. KOSOWSKI, D. PAJAK. *Collision-Free Network Exploration*, in "LATIN - 11th Latin American Theoretical INformatics Symposium", Montevideo, Uruguay, Lecture Notes in Computer Science, Springer, March 2014, vol. 8392, pp. 342-354 [*DOI :* 10.1007/978-3-642-54423-1_30], https://hal.inria.fr/hal-00736276

[17] S. DAS, D. DERENIOWSKI, A. KOSOWSKI, P. UZNANSKI. *Rendezvous of Distance-aware Mobile Agents in Unknown Graphs*, in "21st International Colloquium on Structural Information and Communication Complexity (SIROCCO 2014)", Takayama, Japan, Lecture Notes in Computer Science, Springer Verlag, July 2014, vol. 8576, pp. 295-310 [*DOI :* 10.1007/978-3-319-09620-9_24], https://hal.inria.fr/hal-00986103

[18] C. DELPORTE-GALLET, H. FAUCONNIER, E. GAFNI, L. LAMPORT. *Allocation adaptative de registres en utilisant un nombre linéaire de registres*, in "ALGOTEL 2014 – 16èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications", Le Bois-Plage-en-Ré, France, June 2014, pp. 1-4, https://hal.archives-ouvertes.fr/hal-00978860

[19] C. DELPORTE-GALLET, H. FAUCONNIER, M. RAYNAL. *An Exercise in Concurrency: From Non-blocking Objects to Fair Objects*, in "International Conference on Network-Based Information Systems(NBiS 2014)", Salerne, Italy, September 2014, https://hal.inria.fr/hal-01100781

[20] C. DELPORTE-GALLET, H. FAUCONNIER, M. RAYNAL. *Fair Synchronization in the Presence of Process Crashes and its Weakest Failure Detector*, in "33h Symposium on Reliable Distributed Systems (SRDS)", Nara, Japan, October 2014, pp. 161-170 [*DOI :* 10.1109/SRDS.2014.18], https://hal.inria.fr/hal-01100780

[21] C. DELPORTE-GALLET, H. FAUCONNIER, H. TRAN-THE. *Leader Election in Rings with Homonyms*, in "NETYS", Marrackech, Morocco, Springer, May 2014, pp. 9-24 [*DOI :* 10.1007/978-3-319-09581-3 2], https://hal.inria.fr/hal-01100776

[22] D. DERENIOWSKI, R. KLASING, A. KOSOWSKI, Ł. KUSZNER. *Rendezvous of Heterogeneous Mobile Agents in Edge-Weighted Networks*, in "21th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2014)", Japan, Lecture Notes in Computer Science, Springer Verlag, July 2014, vol. 8576, pp. 311–326, https://hal.archives-ouvertes.fr/hal-01068911

[23] D. DERENIOWSKI, A. KOSOWSKI, D. PAJAK, P. UZNANSKI. *Bounds on the Cover Time of Parallel Rotor Walks*, in "STACS 2014", Lyon, France, March 2014, pp. 263–275 [*DOI :* 10.4230/LIPICS.STACS.2014.263], https://hal.archives-ouvertes.fr/hal-00865065

[24] F. DURAND, F. MATHIEU, L. NOIRIE. *Élection d'un chemin dans un réseau : étude de la manipulabilité*, in "ALGOTEL 2014 – 16èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications", Le Bois-Plage-en-Ré, France, June 2014, pp. 1-4, https://hal.inria.fr/hal-00986050

[25] F. DURAND, F. MATHIEU, L. NOIRIE. *Élection du Best Paper AlgoTel 2012 : étude de la manipulabilité*, in "ALGOTEL 2014 – 16èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications", Le Bois-Plage-en-Ré, France, June 2014, pp. 1-4, https://hal.inria.fr/hal-00986060

[26] O. FEINERMAN, H. BERNHARD, A. KORMAN. *Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication*, in "Proceedings of the 2014 ACM symposium on Principles of distributed computing", Paris, France, July 2014 [*DOI :* 10.1145/2611462.2611469], https://hal.inria.fr/hal-01093278

[27] P. FRAIGNIAUD, E. GAFNI, S. RAJSBAUM, M. ROY. *Automatically Adjusting Concurrency to the Level of Synchrony*, in "DISC 2014 - 28th International Symposium on Distributed Computing", Austin, United States, F. KUHN (editor), LNCS - Lecture Notes in Computer Science, Springer, October 2014, vol. 8784, pp. 1-15, ISBN 978-3-662-45173-1 [*DOI :* 10.1007/978-3-662-45174-8_1], https://hal.inria.fr/hal-01102133

[28] P. FRAIGNIAUD, S. RAJSBAUM, M. ROY, C. TRAVERS. *The Opinion Number of Set-Agreement*, in "OPODIS 2014 - 18th International Conference on Principles of Distributed Systems", Cortina d'Ampezzo, Italy, M. K. AGUILERA, L. QUERZONI, M. SHAPIRO (editors), LNCS - Lecture Notes in Computer Science, Springer , December 2014, vol. 8878, pp. 155-170, https://hal.inria.fr/hal-01102118

[29] T. D. HUYNH, F. MATHIEU, L. VIENNOT. *LiveRank : comment faire du neuf avec du vieux ?*, in "ALGOTEL 2014 – 16èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications", Le Bois-Plage-en-Ré, France, June 2014, pp. 1-4, https://hal.inria.fr/hal-00986031

[30] T. D. HUYNH, F. MATHIEU, L. VIENNOT. *LiveRank: How to Refresh Old Crawls*, in "Algorithms and Models for the Web Graph - 11th International Workshop (WAW 2014)", Beijing, China, December 2014, pp. 148 - 160 [*DOI :* 10.1007/978-3-319-13123-8_12], https://hal.inria.fr/hal-01093188

[31] T. D. HUYNH, C. SHUE CHEN, H. SIU-WAI. *Exploiting User Movement for Position Detection* , in "IEEE Consumer Communications and Networking Conference", Las Vegas, United States, IEEE, January 2015, 6 p. , https://hal.archives-ouvertes.fr/hal-01090484

[32] A. KORMAN, O. FEINERMAN, R. YOAV. *Fast Rendezvous on a Cycle by Agents with Different Speeds*, in "ICDCN", Coimbatore, India, January 2014 [*DOI :* 10.1007/978-3-642-45249-9_1], https://hal.inria.fr/hal-01093272

[33] A. KOSOWSKI, D. PAJAK. *Does Adding More Agents Make a Difference? A Case Study of Cover Time for the Rotor-Router*, in "ICALP - 41st International Colloquium on Automata, Languages, and Programming", Copenhagen, Denmark, Lecture Notes in Computer Science, Springer Verlag, July 2014, vol. 8573, pp. 544-555 [*DOI :* 10.1007/978-3-662-43951-7_46], https://hal.inria.fr/hal-00950743

[34] D. PERINO, M. VARVELLO, L. LINGUAGLOSSA, R. LAUFER, R. BOISLAIGUE. *Caesar: A Content Router for High-Speed Forwarding on Content Names*, in "ACM/IEEE Symposium on Architectures for Networking and Communications Systems", Marina del Rey - Los Angeles, United States, October 2014 [*DOI :* 10.1145/2658260.2658267], https://hal.inria.fr/hal-01101460

Conferences without Proceedings

[35] F. DURAND, B. KLOECKNER, F. MATHIEU, L. NOIRIE. *Geometry on the Utility Space*, in "The 12th Meeting of the Society for Social Choice and Welfare (SSCW 2014)", Boston, United States, June 2014, https://hal.archives-ouvertes.fr/hal-01096018

Research Reports

[36] C. DELPORTE-GALLET, H. FAUCONNIER. *Asynchronous Consensus with Bounded Memory*, January 2014, https://hal.archives-ouvertes.fr/hal-00988590

[37] C. DELPORTE-GALLET, H. FAUCONNIER, S. TOUEG. *A separation of (n -1)-consensus and n-consensus in read-write shared-memory systems*, May 2014, https://hal.archives-ouvertes.fr/hal-01023584

[38] F. DURAND, F. MATHIEU, L. NOIRIE. *Making a voting system depend only on orders of preference reduces its manipulability rate*, June 2014, 26 p. , https://hal.inria.fr/hal-01009136

[39] F. DURAND, F. MATHIEU, L. NOIRIE. *Making most voting systems meet the Condorcet criterion reduces their manipulability*, June 2014, 27 p. , https://hal.inria.fr/hal-01009134

[40] M. MARKIEWICZ, A. KOSOWSKI, T. TYLEC, J. PYKACZ, C. GAVOILLE. *Static Quantum Games Revisited*, June 2014, https://hal.archives-ouvertes.fr/hal-00999012

Other Publications

[41] P. BERENBRINK, R. KLASING, A. KOSOWSKI, F. MALLMANN-TRENN, P. UZNANSKI. *Improved Analysis of Deterministic Load-Balancing Schemes*, July 2014, minor corrections; updated literature overview, https://hal.inria.fr/hal-00979691

[42] J. CHALOPIN, S. DAS, P. GAWRYCHOWSKI, A. KOSOWSKI, A. LABOUREL, P. UZNANSKI. *Lock-in Problem for Parallel Rotor-router Walks*, July 2014, https://hal.inria.fr/hal-01021930

[43] D. DERENIOWSKI, R. KLASING, A. KOSOWSKI, L. KUSZNER. *Rendezvous of Heterogeneous Mobile Agents in Edge-weighted Networks*, June 2014, https://hal.archives-ouvertes.fr/hal-01003010

[44] F. DURAND, F. MATHIEU, L. NOIRIE. *Reducing Manipulability*, June 2014, Fifth International Workshop on Computational Social Choice (ComSoC - 14), https://hal.archives-ouvertes.fr/hal-01095992

[45] D. PERINO, M. GALLO, R. BOISLAIGUE, L. LINGUAGLOSSA, M. VARVELLO, G. CAROFIGLIO, L. MUSCARIELLO, B. H. ZIED. *A High Speed Information-Centric Network in a Mobile Backhaul Setting*, September 2014, ACM Conference on Information-Centric Networking [*DOI :* 10.1145/2660129.2660134], https://hal.inria.fr/hal-01101463