RESEARCH CENTER
Lille - Nord Europe

FIELD

Activity Report 2019

Section Scientific Foundations

Edition: 2020-03-21

BONUS Project-Team

3. Research Program

3.1. Decomposition-based Optimization

Given the large scale of the targeted optimization problems in terms of the number of variables and objectives, their decomposition into simplified and loosely coupled or independent subproblems is essential to raise the challenge of scalability. The first line of research is to *investigate the decomposition approach in the two spaces and their combination, as well as their implementation on ultra-scale architectures*. The motivation of the decomposition is twofold: first, the decomposition allows the parallel resolution of the resulting subproblems on ultra-scale architectures. Here also several issues will be addressed: the definition of the subproblems, their coding to allow their efficient communication and storage (checkpointing), their assignment to processing cores etc. Second, decomposition is necessary for solving large problems that cannot be solved (efficiently) using traditional algorithms. Indeed, for instance with the popular NSGA-II algorithm the number of non-dominated solutions [0] increases drastically with the number of objectives leading to a very slow convergence to the Pareto Front [0]. Therefore, decomposition-based techniques are gaining a growing interest. The objective of BONUS is to *investigate various decomposition schema and cooperation protocols between the subproblems* resulting from the decomposition to generate efficiently global solutions of good quality. Several challenges have to be addressed: (1) how to define the subproblems (decomposition strategy), (2) how to solve them to generate local solutions (local rules), and (3) how to combine these latter with those generated by other subproblems and how to generate global solutions (cooperation mechanism), and (4) how to combine decomposition strategies in more than one space (hybridization strategy)? These challenges, which are in the line with the CIS Task Force [0] on decomposition will be addressed in the decision as well as in the objective space.

The *decomposition in the decision space* can be performed following different ways according to the problem at hand. Two major categories of decomposition techniques can be distinguished: the first one consists in *breaking down the high-dimensional decision vector* into lower-dimensional and easier-to-optimize blocks of variables. The major issue is how to define the subproblems (blocks of variables) and their cooperation protocol: randomly *vs.* using some learning (e.g. separability analysis), statically *vs.* adaptively etc. *The decomposition in the decision space can also be guided by the type of variables i.e. discrete vs. continuous*. The discrete and continuous parts are optimized separately using cooperative hybrid algorithms [48]. *The major issue of this kind of decomposition is the presence of categorial variables in the discrete part [44]. The* BONUS *team is addressing this issue, rarely investigated in the literature*, within the context of vehicle aerospace engineering design. The second category consists in the *decomposition according to the ranges of the decision variables*. For continuous problems, the idea consists in iteratively subdividing the search (e.g. design) space into subspaces (hyper-rectangles, intervals etc.) and select those that are most likely to produce the lowest objective function value. *Existing approaches meet increasing difficulty with an increasing number of variables and are often applied to low-dimensional problems. We are investigating this scalability challenge* (e.g. [10]). *For discrete problems, the major challenge is to find a coding (mapping) of the search space to a decomposable entity*. We have proposed an interval-based coding of the permutation space for solving big permutation problems. The approach opens perspectives we are investigating [7], in terms of ultra-scale parallelization, application to multi-permutation problems and hybridization with metaheuristics.

[0] A solution x dominates another solution y if x is better than y for all objectives and there exists at least one objective for which x is strictly better than y.

[0] The Pareto Front is the set of non-dominated solutions.

[0] IEEE CIS Task Force, created in 2017 on Decomposition-based Techniques in Evolutionary Computation.

The *decomposition in the objective space* consists in breaking down an original Many-objective problem (MaOP) into a set of cooperative single-objective subproblems (SOPs). The decomposition strategy requires the careful definition of a scalarizing (aggregation) function and its weighting vectors (each of them corresponds to a separate SOP) to guide the search process towards the best regions. Several scalarizing functions have been proposed in the literature including weighted sum, weighted Tchebycheff, vector angle distance scaling etc. These functions are widely used but they have their limitations. For instance, using weighted Tchebycheff might do harm diversity maintenance and weighted sum is inefficient when it comes to deal with nonconvex Pareto Fronts [40]. Defining a scalarizing function well-suited to the MaOP at hand is therefore a difficult and still an open question being investigated in BONUS [6], [5]. Studying/defining various functions and in-depth analyzing them to better understand the differences between them is required. Regarding the weighting vectors that determine the search direction, their efficient setting is also a key and open issue. They dramatically affect in particular the diversity performance. Their setting rises two main issues: how to determine their number according to the available computational ressources? when (statically or adaptively) and how to determine their values? *Weight adaptation is one of our main concerns that we are addressing especially from a distributed perspective.* They correspond to the main scientific objectives targeted by our bilateral ANR-RGC BigMO project with City University (Hong Kong). The other challenges pointed out in the beginning of this section concern the way to solve locally the SOPs resulting from the decomposition of a MaOP and the mechanism used for their cooperation to generate global solutions. To deal with these challenges, our approach is to design the decomposition strategy and cooperation mechanism keeping in mind the parallel and/or distributed solving of the SOPs. Indeed, we favor the local neighborhood-based mating selection and replacement to minimize the network communication cost while allowing an effective resolution [5]. The major issues here are how to define the neighborhood of a subproblem and how to cooperatively update the best-known solution of each subproblem and its neighbors.

To sum up, the objective of the BONUS *team is to come up with scalable decomposition-based approaches in the decision and objective spaces. In the decision space, a particular focus will be put on high dimensionality and mixed-continuous variables which have received little interest in the literature. We will particularly continue to investigate at larger scales using ultra-scale computing the interval-based (discrete) and fractal-based (continuous) approaches. We will also deal with the rarely addressed challenge of mixed-continuous including categorial variables (collaboration with ONERA). In the objective space, we will investigate parallel ultra-scale decomposition-based many-objective optimization with ML-based adaptive building of scalarizing functions. A particular focus will be put on the state-of-the-art MOEA/D algorithm. This challenge is rarely addressed in the literature which motivated the collaboration with the designer of MOEA/D (bilateral ANR-RGC BigMO project with City University, Hong Kong). Finally, the joint decision-objective decomposition, which is still in its infancy [50], is another challenge of major interest.*

3.2. Machine Learning-assisted Optimization

The Machine Learning (ML) approach based on metamodels (or surrogates) is commonly used, and also adopted in BONUS, to assist optimization in tackling BOPs characterized by time-demanding objective functions. The second line of research of BONUS is focused on ML-aided optimization to raise the challenge of expensive functions of BOPs using surrogates but also to assist the two other research lines (decomposition-based and ultra-scale optimization) in dealing with the other challenges (high dimensionality and scalability). Several issues have been identified to make efficient and effective surrogate-assisted optimization. First, infill criteria have to be carefully defined to adaptively select the adequate sample points (in terms of surrogate precision and solution quality). The challenge is to find the best trade-off between exploration and exploitation to efficiently refine the surrogate and guide the optimization process toward the best solutions. The most popular infill criterion is probably the *Expected Improvement* (EI) [43] which is based on the expected values of sample points but also and importantly on their variance. This latter is inherently determined in the kriging model, this is why it is used in the state-of-the-art *efficient global optimization* (EGO) algorithm [43]. However, such crucial information is not provided in all surrogate models (e.g. Artificial Neural Networks) and needs to be derived. In BONUS, we are currently investigating this issue. Second, it is known that surrogates allow one to reduce the computational burden for solving BOPs with time-demanding function(s). However,

using parallel computing as a complementary way is often recommended and cited as a perspective in the conclusions of related publications. Nevertheless, *despite being of critical importance parallel surrogate-assisted optimization is weakly addressed in the literature*. For instance, in the introduction of the survey proposed in [42] it is warned that because the area is not mature yet the paper is more focused on the potential of the surveyed approaches than on their relative efficiency. *Parallel computing is required at different levels that we are investigating*.

Another issue with surrogate-assisted optimization is related to high dimensionality in decision as well as in objective space: it is often applied to low-dimensional problems. *The joint use of decomposition, surrogates and massive parallelism is an efficient approach to deal with high dimensionality. This approach adopted in* BONUS *has received little effort in the literature.* In BONUS, we are considering a generic framework in order to enable a flexible coupling of existing surrogate models within the state-of-the-art decomposition-based algorithm MOEA/D. This is a first step in leveraging the applicability of efficient global optimization into the multi-objective setting through parallel decomposition. Another issue which is a consequence of high dimensionality is the mixed (discrete-continuous) nature of decision variables which is frequent in real-world applications (e.g. engineering design). *While surrogate-assisted optimization is widely applied in the continuous setting it is rarely addressed in the literature in the discrete-continuous framework.* In [44], we have identified different ways to deal with this issue that we are investigating. Non-stationary functions frequent in real-world applications (see Section 4.1) is another major issue we are addressing using the concept of deep GP.

Finally, as quoted in the beginning of this section, ML-assisted optimization is mainly used to deal with BOPs with expensive functions but it will also be investigated for other optimization tasks. Indeed, ML will be useful to assist the decomposition process. In the decision space, it will help to perform the separability analysis (understanding of the interactions between variables) to decompose the vector of variables. In the objective space, ML will be useful to assist a decomposition-based many-objective algorithm in dynamically selecting a scalarizing function or updating the weighting vectors according to their performances in the previous steps of the optimization process [5]. Such a data-driven ML methodology would allow us to understand what makes a problem difficult or an optimization approach efficient, to predict the algorithm performance [4], to select the most appropriate algorithm configuration [8], and to adapt and improve the algorithm design for unknown optimization domains and instances. Such an autonomous optimization approach would adaptively adjust its internal mechanisms in order to tackle cross-domain BOPs.

In a nutshell, to deal with expensive optimization the BONUS *team will investigate the surrogate-based ML approach with the objective to efficiently integrate surrogates in the optimization process. The focus will especially be put on high dimensionality (e.g. using decomposition) with mixed discrete-continuous variables which is rarely investigated. The kriging metamodel (Gaussian Process or GP) will be considered in particular for engineering design (for more reliability) addressing the above issues and other major ones including mainly non stationarity (using emerging deep GP) and ultra-scale parallelization (highly needed by the community). Indeed, a lot of work has been reported on deep neural networks (deep learning) surrogates but not on the others including (deep) GP. On the other hand, ML will be used to assist decomposition: importance/interaction between variables in the decision space, dynamic building (selection of scalarizing functions, weight update etc.) of scalarizing functions in the objective space etc.*

3.3. Ultra-scale Optimization

The third line of our research program that accentuates our difference from other (project-)teams of the related Inria scientific theme is the ultra-scale optimization. *This research line is complementary to the two others, which are sources of massive parallelism* and with which it should be combined to solve BOPs. Indeed, ultra-scale computing is necessary for the effective resolution of the large amount of subproblems generated by decomposition of BOPs, parallel evaluation of simulation-based fitness and metamodels etc. These sources of parallelism are attractive for solving BOPs and are natural candidates for ultra-scale supercomputers [0].

[0] In the context of BONUS, supercomputers are composed of several massively parallel processing nodes (inter-node parallelism) including multi-core processors and GPUs (intra-node parallelism).

However, their efficient use raises a big challenge consisting in managing efficiently a massive amount of irregular tasks on supercomputers with multiple levels of parallelism and heterogeneous computing resources (GPU, multi-core CPU with various architectures) and networks. Raising such challenge requires to tackle three major issues, scalability, heterogeneity and fault-tolerance, discussed in the following.

The *scalability* issue requires, on the one hand, the definition of scalable data structures for efficient storage and management of the tremendous amount of subproblems generated by decomposition [46]. On the other hand, achieving extreme scalability requires also the optimization of communications (in number of messages, their size and scope) especially at the inter-node level. For that, we target the design of asynchronous locality-aware algorithms as we did in [41], [49]. In addition, efficient mechanisms are needed for granularity management and coding of the work units stored and communicated during the resolution process.

Heterogeneity means harnessing various resources including multi-core processors within different architectures and GPU devices. The challenge is therefore to design and implement hybrid optimization algorithms taking into account the difference in computational power between the various resources as well as the resource-specific issues. On the one hand, to deal with the heterogeneity in terms of computational power, we adopt in BONUS the dynamic load balancing approach based on the Work Stealing (WS) asynchronous paradigm [0] at the inter-node as well as at the intra-node level. We have already investigated such approach, with various victim selection and work sharing strategies in [49], [7]. On the other hand, hardware resource specific-level optimization mechanisms are required to deal with related issues such as thread divergence and memory optimization on GPU, data sharing and synchronization, cache locality, and vectorization on multi-core processors etc. These issues have been considered separately in the literature including our works [9], [1]. Indeed, in most of existing works related to GPU-accelerated optimization only a single CPU core is used. This leads to a huge resource wasting especially with the increase of the number of processing cores integrated into modern processors. Using jointly the two components raises additional issues including data and work partitioning, the optimization of CPU-GPU data transfers etc.

Another issue the scalability induces is the *increasing probability of failures* in modern supercomputers [47]. Indeed, with the increase of their size to millions of processing cores their Mean-Time Between Failures (MTBF) tends to be shorter and shorter [45]. Failures may have different sources including hardware and software faults, silent errors etc. In our context, we consider failures leading to the loss of work unit(s) being processed by some thread(s) during the resolution process. The major issue, which is particularly critical in exact optimization, is how to recover the failed work units to ensure a reliable execution. Such issue is tackled in the literature using different approaches: algorithm-based fault tolerance, checkpoint/restart (CR), message logging and redundancy. The CR approach can be system-level, library/user-level or application-level. Thanks to its efficiency in terms of memory footprint, adopted in BONUS [2], the application-level approach is commonly and widely used in the literature. This approach raises several issues mainly: (1) which critical information defines the state of the work units and allows to resume properly their execution? (2) when, where and how (using which data structures) to store it efficiently? (3) how to deal with the two other issues: scalability and heterogeneity?
The last but not least major issue which is another roadblock to exascale is the programming of massive-scale applications for modern supercomputers. *On the path to exascale, we will investigate the programming environments and execution supports able to deal with exascale challenges: large numbers of threads, heterogeneous resources etc.* Various exascale programming approaches are being investigated by the parallel computing community and HPC builders: extending existing programming languages (e.g. DSL-C++) and environments/libraries (MPI+X etc.), proposing new solutions including mainly Partitioned Global Address Space (PGAS)-based environments (Chapel, UPC, X10 etc.). It is worth noting here that our objective is not to develop a programming environment nor a runtime support for exascale computing. Instead, we aim to collaborate with the research teams (inside or outside Inria) having such objective.

[0] A WS mechanism is mainly defined by two components: a victim selection strategy which selects the processing core to be stolen and a work sharing policy which determines the part and amount of the work unit to be given to the thief upon WS request.

To sum up, we put the focus on the design and implementation of efficient big optimization algorithms dealing jointly (uncommon in parallel optimization) with the major issues of ultra-scale computing mainly the scalability up to millions of cores using scalable data structures and asynchronous locality-aware work stealing, heterogeneity addressing the multi-core and GPU-specific issues and those related to their combination, and scalable GPU-aware fault tolerance. A strong effort will be devoted to this latter challenge, for the first time to the best of our knowledge, using application-level checkpoint/restart approach to deal with failures.

INOCS Project-Team

3. Research Program

3.1. Introduction

An optimization problem consists in finding a best solution from a set of feasible solutions. Such a problem can be typically modeled as a mathematical program in which decision variables must:

1. satisfy a set of constraints that translate the feasibility of the solution and
2. optimize some (or several) objective function(s). Optimization problems are usually classified according to types of decision to be taken into strategic, tactical and operational problems.

We consider that an optimization problem presents a complex structure when it involves decisions of different types/nature (i.e. strategic, tactical or operational), and/or presenting some hierarchical leader-follower structure. The set of constraints may usually be partitioned into global constraints linking variables associated with the different types/nature of decision and constraints involving each type of variables separately. Optimization problems with a complex structure lead to extremely challenging problems since a global optimum with respect to the whole sets of decision variables and of constraints must be determined.

Significant progresses have been made in optimization to solve academic problems. Nowadays large-scale instances of some NP-Hard problems are routinely solved to optimality. *Our vision within INOCS is to make the same advances while addressing CS optimization problems*. To achieve this goal we aim to develop global solution approaches at the opposite of the current trend. INOCS team members have already proposed some successful methods following this research lines to model and solve CS problems (e.g. ANR project RESPET, Brotcorne *et al.* 2011, 2012, Gendron *et al.* 2009, Strack *et al.* 2009). However, these are preliminary attempts and a number of challenges regarding modeling and methodological issues have still to be met.

3.2. Modeling problems with complex structures

A classical optimization problem can be formulated as follows:

$$\begin{array}{ll} \min & f(x) \\ s.t. & x \in X. \end{array} \tag{1}$$

In this problem, X is the set of feasible solutions. Typically, in mathematical programming, X is defined by a set of constraints. x may be also limited to non-negative integer values.

INOCS team plan to address optimization problem where two types of decision are addressed jointly and are interrelated. More precisely, let us assume that variables x and y are associated with these decisions. A generic model for CS problems is the following:

$$\begin{array}{ll} \min & g(x,y) \\ s.t. & x \in X, \\ & (x,y) \in XY, \\ & y \in Y(x). \end{array} \tag{2}$$

In this model, X is the set of feasible values for x. XY is the set of feasible values for x and y jointly. This set is typically modeled through linking constraints. Last, $Y(x)$ is the set of feasible values for y for a given x. In INOCS, we do not assume that $Y(x)$ has any properties.

The INOCS team plans to model optimization CS problems according to three types of optimization paradigms: large scale complex structures optimization, bilevel optimization and robust/stochastic optimization. These paradigms instantiate specific variants of the generic model.

Large scale complex structures optimization problems can be formulated through the simplest variant of the generic model given above. In this case, it is assumed that $Y(x)$ does not depend on x. In such models, X and Y are associated with constraints on x and on y, XY are the linking constraints. x and y can take continuous or integer values. Note that all the problem data are deterministically known.

Bilevel programs allow the modeling of situations in which a decision-maker, hereafter the leader, optimizes his objective by taking explicitly into account the response of another decision maker or set of decision makers (the follower) to his/her decisions. Bilevel programs are closely related to Stackelberg (leader-follower) games as well as to the principal-agent paradigm in economics. In other words, bilevel programs can be considered as demand-offer equilibrium models where the demand is the result of another mathematical problem. Bilevel problems can be formulated through the generic CS model when $Y(x)$ corresponds to the optimal solutions of a mathematical program defined for a given x, i.e. $Y(x) = \arg\min \{h(x,y) | y \in Y_2, (x,y) \in XY_2\}$ where Y_2 is defined by a set of constraints on y, and XY_2 is associated with the linking constraints.

In robust/stochastic optimization, it is assumed that the data related to a problem are subject to uncertainty. In stochastic optimization, probability distributions governing the data are known, and the objective function involves mathematical expectation(s). In robust optimization, uncertain data take value within specified sets, and the function to optimize is formulated in terms of a min-max objective typically (the solution must be optimal for the worst-case scenario). A standard modeling of uncertainty on data is obtained by defining a set of possible scenarios that can be described explicitly or implicitly. In stochastic optimization, in addition, a probability of occurrence is associated with each scenario and the expected objective value is optimized.

3.3. Solving problems with complex structures

Standard solution methods developed for CS problems solve independent sub-problems associated with each type of variables without explicitly integrating their interactions or integrating them iteratively in a heuristic way. However these subproblems are intrinsically linked and should be addressed jointly. In *mathematicaloptimization* a classical approach is to approximate the convex hull of the integer solutions of the model by its linear relaxation. The main solution methods are (1) polyhedral solution methods which strengthen this linear relaxation by adding valid inequalities, (2) decomposition solution methods (Dantzig Wolfe, Lagrangian Relaxation, Benders decomposition) which aim to obtain a better approximation and solve it by generating extreme points/rays. Main challenges are (1) the analysis of the strength of the cuts and their separations for polyhedral solution methods, (2) the decomposition schemes and (3) the extreme points/rays generations for the decomposition solution methods.

The main difficulty in solving *bilevel problems* is due to their non convexity and non differentiability. Even linear bilevel programs, where all functions involved are affine, are computationally challenging despite their apparent simplicity. Up to now, much research has been devoted to bilevel problems with linear or convex follower problems. In this case, the problem can be reformulated as a single-level program involving complementarity constraints, exemplifying the dual nature, continuous and combinatorial, of bilevel programs.

MEPHYSTO Team

3. Research Program

3.1. Time asymptotics: Stationary states, solitons, and stability issues

The team investigates the existence of solitons and their link with the global dynamical behavior for nonlocal problems such as that of the Gross–Pitaevskii (GP) equation which arises in models of dipolar gases. These models, in general, also introduce nonzero boundary conditions which constitute an additional theoretical and numerical challenge. Numerous results are proved for local problems, and numerical simulations allow to verify and illustrate them, as well as making a link with physics. However, most fundamental questions are still open at the moment for nonlocal problems.

The nonlinear Schrödinger (NLS) equation finds applications in numerous fields of physics. We concentrate, in a continued collaboration with our colleagues from the physics department (PhLAM) of the Université de Lille (UdL), in the framework of the Laboratoire d'Excellence CEMPI, on its applications in nonlinear optics and cold atom physics. Issues of orbital stability and modulational instability are central here.

Another typical example of problems that the team wishes to address concerns the Landau–Lifshitz (LL) equation, which describes the dynamics of the spin in ferromagnetic materials. This equation is a fundamental model in the magnetic recording industry [37] and solitons in magnetic media are of particular interest as a mechanism for data storage or information transfer [38]. It is a quasilinear PDE involving a function that takes values on the unit sphere $\mathbb{S}^2$ of $\mathbb{R}^3$. Using the stereographic projection, it can be seen as a quasilinear Schrödinger equation and the questions about the solitons, their dynamics and potential blow-up of solutions evoked above are also relevant in this context. This equation is less understood than the NLS equation: even the Cauchy theory is not completely done [36], [35]. In particular, the geometry of the target sphere imposes nonvanishing boundary conditions; even in dimension one, there are kink-type solitons having different limits at $\pm\infty$.

3.2. Derivation of macroscopic laws from microscopic dynamics

The team investigates, from a microscopic viewpoint, the dynamical mechanism at play in the phenomenon of relaxation towards thermal equilibrium for large systems of interacting particles. For instance, a first step consists in giving a rigorous proof of the fact that a particle repeatedly scattered by random obstacles through a Hamiltonian scattering process will eventually reach thermal equilibrium, thereby completing previous work in this direction by the team. As a second step, similar models as the ones considered classically will be defined and analysed in the quantum mechanical setting, and more particularly in the setting of quantum optics.

Another challenging problem is to understand the interaction of large systems with the boundaries, which is responsible for most energy exchanges (forcing and dissipation), even though it is concentrated in very thin layers. The presence of boundary conditions to evolution equations sometimes lacks understanding from a physical and mathematical point of view. In order to legitimate the choice done at the macroscopic level of the mathematical definition of the boundary conditions, we investigate systems of atoms (precisely chains of oscillators) with different local microscopic defects. We apply our recent techniques to understand how anomalous (in particular fractional) diffusive systems interact with the boundaries. For instance, the powerful tool given by Wigner functions that we already used has been successfully applied to the derivation of anomalous behaviors in open systems (for instance in [7]). The next step consists in developing an extension of that tool to deal with bounded systems provided with fixed boundaries. We also intend to derive anomalous diffusion by adding long range interactions to diffusive models. There are very few rigorous results in this direction. Finally, we aim at obtaining from a microscopic description the fractional porous medium equation (FPME), a nonlinear variation of the fractional diffusion equation, involving the fractional Laplacian instead of the usual one. Its rigorous study carries out many mathematical difficulties in treating at the same time the

nonlinearity and fractional diffusion. We want to make PDE theorists and probabilists work together, in order to take advantage of the analytical results which went far ahead and are more advanced than the statistical physics theory.

3.3. Numerical methods: analysis and simulations

The team addresses both questions of precision and numerical cost of the schemes for the numerical integration of nonlinear evolution PDEs, such as the NLS equation. In particular, we aim at developing, studying and implementing numerical schemes with high order that are more efficient for these problems. We also want to contribute to the design and analysis of schemes with appropriate qualitative properties. These properties may as well be "asymptotic preserving" properties, energy-preserving properties, or convergence to an equilibrium properties. Other numerical goals of the team include the numerical simulation of standing waves of nonlinear nonlocal GP equations. We also keep on developing numerical methods to efficiently simulate and illustrate theoretical results on instability, in particular in the context of the modulational instability in optical fibers, where we study the influence of randomness in the physical parameters of the fibers.

The team also designs simulation methods to estimate the accuracy of the physical description via microscopic systems, by computing precisely the rate of convergence as the system size goes to infinity. One method under investigation is related to cloning algorithms, which were introduced very recently and turn out to be essential in molecular simulation.

MODAL Project-Team

3. Research Program

3.1. Research axis 1: Unsupervised learning

Scientific locks related to unsupervised learning are numerous, concerning the clustering outcome validity, the ability to manage different kinds of data, the missing data questioning, the dimensionality of the data set,... Many of them are addressed by the team, leading to publication achievements, often with a specific package delivery (sometimes upgraded as a software or even as a platform grouping several software). Because of the variety of the scope, it involves nearly all the permanent team members, often with PhD students and some engineers. The related works are always embedded inside a probabilistic framework, typically model-based approaches but also model-free ones like PAC-Bayes (PAC stands for Probably Approximately Correct), because such a mathematical environment offers both a well-posed problem and a rigorous answer.

3.2. Research axis 2: Performance assessment

One main concern of the Modal team is to provide theoretical justifications on the procedures which are designed. Such guarantees are important to avoid misleading conclusions resulting from any unsuitable use. For example, one ingredient in proving these guarantees is the use of the PAC framework, leading to finite-sample concentration inequalities. More precisely, contributions to PAC learning rely on the classical empirical process theory and the PAC-Bayesian theory. The Modal team exploits such non-asymptotic tools to analyze the performance of iterative algorithms (such as gradient descent), cross-validation estimators, online change-point detection procedures, ranking algorithms, matrix factorization techniques and clustering methods, for instance. The team also develops some expertise on the formal dynamic study of algorithms related to mixture models (important models used in the previous unsupervised setting), like degeneracy for EM algorithm or also label switching for Gibbs algorithm.

3.3. Research axis 3: Functional data

Mainly due to technological advances, functional data are more and more widespread in many application domains. Functional data analysis (FDA) is concerned with the modeling of data, such as curves, shapes, images or a more complex mathematical object, though as smooth realizations of a stochastic process (an infinite dimensional data object valued in a space of eventually infinite dimension; space of squared integrable functions,...). Time series are an emblematic example even if it should not be limited to them (spectral data, spatial data,...). Basically, FDA considers that data correspond to realizations of stochastic processes, usually assumed to be in a metric, semi-metric, Hilbert or Banach space. One may consider, functional independent or dependent (in time or space) data objects of different types (qualitative, quantitative, ordinal, multivariate, time-dependent, spatial-dependent,...). The last decade saw a dynamic literature on parametric or non-parametric FDA approaches for different types of data and applications to various domains, such as principal component analysis, clustering, regression and prediction.

3.4. Research axis 4: Applications motivating research

The fourth axis consists in translating real application issues into statistical problems raising new (academic) challenges for models developed in Modal team. Cifre Phds in industry and interdisciplinary projects with research teams in Health and Biology are at the core of this objective. The main originality of this objective lies in the use of statistics with complex data, including in particular ultra-high dimension problems. We focus on real applications which cannot be solved by classical data analysis.

RAPSODI Project-Team

3. Research Program

3.1. Design and analysis of structure-preserving schemes

3.1.1. Numerical analysis of nonlinear numerical methods

Up to now, the numerical methods dedicated to degenerate parabolic problems that the mathematicians are able to analyze almost all rely on the use of mathematical transformations (like e.g. the Kirchhoff's transform). It forbids the extension of the analysis to complex realistic models. The methods used in the industrial codes for solving such complex problems rely on the use of what we call NNM, i.e., on methods that preserve all the nonlinearities of the problem without reducing them thanks to artificial mathematical transforms. Our aim is to take advantage of the recent breakthrough proposed by C. Cancès & C. Guichard [83], [4] to develop efficient new numerical methods with a full numerical analysis (stability, convergence, error estimates, robustness w.r.t. physical parameters,...).

3.1.2. Design and analysis of asymptotic-preserving schemes

There has been an extensive effort in the recent years to develop numerical methods for diffusion equations that are robust with respect to heterogeneities, anisotropy, and the mesh (see for instance [98] for an extensive discussion on such methods). On the other hand, the understanding of the role of nonlinear stability properties in the asymptotic behaviors of dissipative systems increased significantly in the last decades (see for instance [85], [110]).

Recently, C. Chainais-Hillairet and co-authors [79], [86] and [87] developed a strategy based on the control of the numerical counterpart of the physical entropy to develop and analyze AP numerical methods. In particular, these methods show great promises for capturing accurately the behavior of the solutions to dissipative problems when some physical parameter is small with respect to the discretization characteristic parameters, or in the long-time asymptotic. Since it requires the use of nonlinear test functions in the analysis, strong restrictions on the physics (isotropic problems) and on the mesh (Cartesian grids, Voronoï boxes...) are required in [79], [86] and [87]. The schemes proposed in [83] and [4] allow to handle nonlinear test functions in the analysis without restrictions on the mesh and on the anisotropy of the problem. Combining the nonlinear schemes *à la* [83] with the methodology of [79], [86], [87] would provide schemes that are robust both with respect to the meshes and to the parameters. Therefore, they would be also robust under adaptive mesh refinement.

3.1.3. Design and stability analysis of numerical methods for low-Mach models

We aim at extending the range of the `NS2DDV-M` software by introducing new physical models, like for instance the low-Mach model, which gives intermediate solutions between the compressible Navier–Stokes model and the incompressible Navier–Stokes one. This model was introduced in [109] as a limiting system which describes combustion processes at low Mach number in a confined region. Within this scope, we will propose a theoretical study for proving the existence of weak solutions for a particular class of models for which the dynamic viscosity of the fluid is a specific function of the density. We will propose also the extension of a combined Finite Volume-Finite Element method, initially developed for the simulation of incompressible and variable density flows, to this class of models.

3.2. Optimizing the computational efficiency

3.2.1. High-order nonlinear numerical methods

The numerical experiments carried out in [83] show that in case of very strong anisotropy, the convergence of the proposed NNM becomes too slow (less than first order). Indeed, the method appears to strongly overestimate the dissipation. In order to make the method more competitive, it is necessary to estimate the dissipation in a more accurate way. Preliminary numerical results show that second order accuracy in space can be achieved in this way. One also aims to obtain (at least) second order accuracy in time without jeopardizing the stability. For many problems, this can be done by using so-called two-step backward differentiation formulas (BDF2) [99].

Concerning the inhomogeneous fluid models, we aim to investigate new methods for the mass equation resolution. Indeed, we aim at increasing the accuracy while maintaining some positivity-like properties and the efficiency for a wide range of physical parameters. To this end, we will consider Residual Distribution schemes, that appear as an alternative to Finite Volume methods. Residual Distribution schemes enjoy very compact stencils. Therefore, their extension from 2D to 3D yield reasonable difficulties. These methods appeared twenty years ago, but recent extensions to unsteady problems [111], [106], with high-order accuracy [66], [65], or for parabolic problems [63], [64] make them very competitive. Relying on these breakthroughs, we aim at designing new Residual Distribution schemes for fluid mixture models with high-order accuracy while preserving the positivity of the solutions.

3.2.2. A posteriori error control

The question of the *a posteriori* error estimators will also have to be addressed in this optimization context. Since the pioneering papers of Babuska and Rheinboldt more than thirty years ago [70], *a posteriori* error estimators have been widely studied. We will take advantage of the huge corresponding bibliography database in order to optimize our numerical results.

For example, we would like to generalize the results we derived for the harmonic magnetodynamic case (e.g. [88] and [89]) to the temporal magnetodynamic one, for which space/time *a posteriori* error estimators have to be developed. A space/time refinement algorithm should consequently be proposed and tested on academic as well as industrial benchmarks.

We also want to develop *a posteriori* estimators for the variable density Navier–Stokes model or some of its variants. To do so, several difficulties have to be tackled: the problem is nonlinear, unsteady, and the numerical method [81], [82] we developed combines features from Finite Elements and Finite Volumes. Fortunately, we do not start from scratch. Some recent references are devoted to the unsteady Navier–Stokes model in the Finite Element context [77], [114]. In the Finite Volume context, recent references deal with unsteady convection-diffusion equations [113], [68], [96] and [84]. We want to adapt some of these results to the variable density Navier–Stokes system, and to be able to design an efficient space-time remeshing algorithm.

3.2.3. Efficient computation of pairwise interactions in large systems of particles

Many systems are modeled as a large number of punctual individuals (N) which interact pairwise which means $N(N-1)/2$ interactions. Such systems are ubiquitous, they are found in chemistry (Van der Waals interaction between atoms), in astrophysics (gravitational interactions between stars, galaxies or galaxy clusters), in biology (flocking behavior of birds, swarming of fishes) or in the description of crowd motions. Building on the special structure of convolution-type of the interactions, the team develops computation methods based on the Non Uniform Fast Fourier Transform [102]. This reduces the $O(N^2)$ naive computational cost of the interactions to $O(N \log N)$, allowing numerical simulations involving millions of individuals.

SEQUEL Project-Team

3. Research Program

3.1. In Short

SEQUEL is primarily grounded on two domains:

- the problem of decision under uncertainty,
- statistical analysis and statistical learning, which provide the general concepts and tools to solve this problem.

To help the reader who is unfamiliar with these questions, we briefly present key ideas below.

3.2. Decision-making Under Uncertainty

The phrase "Decision under uncertainty" refers to the problem of taking decisions when we do not have a full knowledge neither of the situation, nor of the consequences of the decisions, as well as when the consequences of decision are non deterministic.

We introduce two specific sub-domains, namely the Markov decision processes which model sequential decision problems, and bandit problems.

3.2.1. Reinforcement Learning

Sequential decision processes occupy the heart of the SEQUEL project; a detailed presentation of this problem may be found in Puterman's book [61].

A Markov Decision Process (MDP) is defined as the tuple $(\mathcal{X}, \mathcal{A}, P, r)$ where $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, P is the probabilistic transition kernel, and $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$ is the reward function. For the sake of simplicity, we assume in this introduction that the state and action spaces are finite. If the current state (at time t) is $x \in \mathcal{X}$ and the chosen action is $a \in \mathcal{A}$, then the Markov assumption means that the transition probability to a new state $x' \in \mathcal{X}$ (at time $t+1$) only depends on (x, a). We write $p(x'|x, a)$ the corresponding transition probability. During a transition $(x, a) \to x'$, a reward $r(x, a, x')$ is incurred.

In the MDP $(\mathcal{X}, \mathcal{A}, P, r)$, each initial state x_0 and action sequence $a_0, a_1, ...$ gives rise to a sequence of states $x_1, x_2, ...$, satisfying $\mathbb{P}(x_{t+1} = x'|x_t = x, a_t = a) = p(x'|x, a)$, and rewards [0]$r_1, r_2, ...$ defined by $r_t = r(x_t, a_t, x_{t+1})$.

The history of the process up to time t is defined to be $H_t = (x_0, a_0, ..., x_{t-1}, a_{t-1}, x_t)$. A policy π is a sequence of functions $\pi_0, \pi_1, ...$, where π_t maps the space of possible histories at time t to the space of probability distributions over the space of actions $\mathcal{A}$. To follow a policy means that, in each time step, we assume that the process history up to time t is $x_0, a_0, ..., x_t$ and the probability of selecting an action a is equal to $\pi_t(x_0, a_0, ..., x_t)(a)$. A policy is called stationary (or Markovian) if π_t depends only on the last visited state. In other words, a policy $\pi = (\pi_0, \pi_1, ...)$ is called stationary if $\pi_t(x_0, a_0, ..., x_t) = \pi_0(x_t)$ holds for all $t \geq 0$. A policy is called deterministic if the probability distribution prescribed by the policy for any history is concentrated on a single action. Otherwise it is called a stochastic policy.

[0]Note that for simplicity, we considered the case of a deterministic reward function, but in many applications, the reward r_t itself is a random variable.

We move from an MD process to an MD problem by formulating the goal of the agent, that is what the sought policy π has to optimize. It is very often formulated as maximizing (or minimizing), in expectation, some functional of the sequence of future rewards. For example, an usual functional is the infinite-time horizon sum of discounted rewards. For a given (stationary) policy π, we define the value function $V^\pi(x)$ of that policy π at a state $x \in \mathcal{X}$ as the expected sum of discounted future rewards given that we state from the initial state x and follow the policy π:

$$V^\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | x_0 = x, \pi\right], \tag{3}$$

where $\mathbb{E}$ is the expectation operator and $\gamma \in (0,1)$ is the discount factor. This value function V^π gives an evaluation of the performance of a given policy π. Other functionals of the sequence of future rewards may be considered, such as the undiscounted reward (see the stochastic shortest path problems [60]) and average reward settings. Note also that, here, we consider the problem of maximizing a reward functional, but a formulation in terms of minimizing some cost or risk functional would be equivalent.

In order to maximize a given functional in a sequential framework, one usually applies Dynamic Programming (DP) [58], which introduces the optimal value function $V^*(x)$, defined as the optimal expected sum of rewards when the agent starts from a state x. We have $V^*(x) = \sup_\pi V^\pi(x)$. Now, let us give two definitions about policies:

- We say that a policy π is optimal, if it attains the optimal values $V^*(x)$ for any state $x \in \mathcal{X}$, *i.e.*, if $V^\pi(x) = V^*(x)$ for all $x \in \mathcal{X}$. Under mild conditions, deterministic stationary optimal policies exist [59]. Such an optimal policy is written π^*.
- We say that a (deterministic stationary) policy π is greedy with respect to (w.r.t.) some function V (defined on $\mathcal{X}$) if, for all $x \in \mathcal{X}$,

$$\pi(x) \in \arg\max_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} p(x'|x,a)\left[r(x,a,x') + \gamma V(x')\right].$$

  where $\arg\max_{a \in \mathcal{A}} f(a)$ is the set of $a \in \mathcal{A}$ that maximizes $f(a)$. For any function V, such a greedy policy always exists because $\mathcal{A}$ is finite.

The goal of Reinforcement Learning (RL), as well as that of dynamic programming, is to design an optimal policy (or a good approximation of it).

The well-known Dynamic Programming equation (also called the Bellman equation) provides a relation between the optimal value function at a state x and the optimal value function at the successors states x' when choosing an optimal action: for all $x \in \mathcal{X}$,

$$V^*(x) = \max_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} p(x'|x,a)\left[r(x,a,x') + \gamma V^*(x')\right]. \tag{4}$$

The benefit of introducing this concept of optimal value function relies on the property that, from the optimal value function V^*, it is easy to derive an optimal behavior by choosing the actions according to a policy greedy w.r.t. V^*. Indeed, we have the property that a policy greedy w.r.t. the optimal value function is an optimal policy:

$$\pi^*(x) \in \arg\max_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} p(x'|x,a)\left[r(x,a,x') + \gamma V^*(x')\right]. \tag{5}$$

In short, we would like to mention that most of the reinforcement learning methods developed so far are built on one (or both) of the two following approaches ([64]):

- Bellman's dynamic programming approach, based on the introduction of the value function. It consists in learning a "good" approximation of the optimal value function, and then using it to derive a greedy policy w.r.t. this approximation. The hope (well justified in several cases) is that the performance V^π of the policy π greedy w.r.t. an approximation V of V^* will be close to optimality. This approximation issue of the optimal value function is one of the major challenges inherent to the reinforcement learning problem. **Approximate dynamic programming** addresses the problem of estimating performance bounds (*e.g.* the loss in performance $||V^* - V^\pi||$ resulting from using a policy π-greedy w.r.t. some approximation V- instead of an optimal policy) in terms of the approximation error $||V^* - V||$ of the optimal value function V^* by V. Approximation theory and Statistical Learning theory provide us with bounds in terms of the number of sample data used to represent the functions, and the capacity and approximation power of the considered function spaces.
- Pontryagin's maximum principle approach, based on sensitivity analysis of the performance measure w.r.t. some control parameters. This approach, also called **direct policy search** in the Reinforcement Learning community aims at directly finding a good feedback control law in a parameterized policy space without trying to approximate the value function. The method consists in estimating the so-called **policy gradient**, *i.e.* the sensitivity of the performance measure (the value function) w.r.t. some parameters of the current policy. The idea being that an optimal control problem is replaced by a parametric optimization problem in the space of parameterized policies. As such, deriving a policy gradient estimate would lead to performing a stochastic gradient method in order to search for a local optimal parametric policy.

Finally, many extensions of the Markov decision processes exist, among which the Partially Observable MDPs (POMDPs) is the case where the current state does not contain all the necessary information required to decide for sure of the best action.

3.2.2. Multi-arm Bandit Theory

Bandit problems illustrate the fundamental difficulty of decision making in the face of uncertainty: A decision maker must choose between what seems to be the best choice ("exploit"), or to test ("explore") some alternative, hoping to discover a choice that beats the current best choice.

The classical example of a bandit problem is deciding what treatment to give each patient in a clinical trial when the effectiveness of the treatments are initially unknown and the patients arrive sequentially. These bandit problems became popular with the seminal paper [62], after which they have found applications in diverse fields, such as control, economics, statistics, or learning theory.

Formally, a K-armed bandit problem ($K \geq 2$) is specified by K real-valued distributions. In each time step a decision maker can select one of the distributions to obtain a sample from it. The samples obtained are considered as rewards. The distributions are initially unknown to the decision maker, whose goal is to maximize the sum of the rewards received, or equivalently, to minimize the regret which is defined as the loss compared to the total payoff that can be achieved given full knowledge of the problem, *i.e.*, when the arm giving the highest expected reward is pulled all the time.

The name "bandit" comes from imagining a gambler playing with K slot machines. The gambler can pull the arm of any of the machines, which produces a random payoff as a result: When arm k is pulled, the random payoff is drawn from the distribution associated to k. Since the payoff distributions are initially unknown, the gambler must use exploratory actions to learn the utility of the individual arms. However, exploration has to be carefully controlled since excessive exploration may lead to unnecessary losses. Hence, to play well, the gambler must carefully balance exploration and exploitation. Auer *et al.* [57] introduced the algorithm UCB (Upper Confidence Bounds) that follows what is now called the "optimism in the face of uncertainty principle". Their algorithm works by computing upper confidence bounds for all the arms and then choosing the arm with the highest such bound. They proved that the expected regret of their algorithm increases at most

at a logarithmic rate with the number of trials, and that the algorithm achieves the smallest possible regret up to some sub-logarithmic factor (for the considered family of distributions).

3.3. Statistical analysis of time series

Many of the problems of machine learning can be seen as extensions of classical problems of mathematical statistics to their (extremely) non-parametric and model-free cases. Other machine learning problems are founded on such statistical problems. Statistical problems of sequential learning are mainly those that are concerned with the analysis of time series. These problems are as follows.

3.3.1. Prediction of Sequences of Structured and Unstructured Data

Given a series of observations $x_1, ..., x_n$ it is required to give forecasts concerning the distribution of the future observations $x_{n+1}, x_{n+2}, \cdots$; in the simplest case, that of the next outcome x_{n+1}. Then x_{n+1} is revealed and the process continues. Different goals can be formulated in this setting. One can either make some assumptions on the probability measure that generates the sequence $x_1, \cdots, x_n, \cdots$, such as that the outcomes are independent and identically distributed (i.i.d.), or that the sequence is a Markov chain, that it is a stationary process, etc. More generally, one can assume that the data is generated by a probability measure that belongs to a certain set $\mathcal{C}$. In these cases the goal is to have the discrepancy between the predicted and the "true" probabilities to go to zero, if possible, with guarantees on the speed of convergence.

Alternatively, rather than making some assumptions on the data, one can change the goal: the predicted probabilities should be asymptotically as good as those given by the best reference predictor from a certain pre-defined set.

Another dimension of complexity in this problem concerns the nature of observations x_i. In the simplest case, they come from a finite space, but already basic applications often require real-valued observations. Moreover, function or even graph-valued observations often arise in practice, in particular in applications concerning Web data. In these settings estimating even simple characteristics of probability distributions of the future outcomes becomes non-trivial, and new learning algorithms for solving these problems are in order.

3.3.2. Hypothesis testing

Given a series of observations of $x_1, \cdots, x_n, \cdots$ generated by some unknown probability measure μ, the problem is to test a certain given hypothesis H_0 about μ, versus a given alternative hypothesis H_1. There are many different examples of this problem. Perhaps the simplest one is testing a simple hypothesis "μ is Bernoulli i.i.d. measure with probability of 0 equals 1/2" versus "μ is Bernoulli i.i.d. with the parameter different from 1/2". More interesting cases include the problems of model verification: for example, testing that μ is a Markov chain, versus that it is a stationary ergodic process but not a Markov chain. In the case when we have not one but several series of observations, we may wish to test the hypothesis that they are independent, or that they are generated by the same distribution. Applications of these problems to a more general class of machine learning tasks include the problem of feature selection, the problem of testing that a certain behavior (such as pulling a certain arm of a bandit, or using a certain policy) is better (in terms of achieving some goal, or collecting some rewards) than another behavior, or than a class of other behaviors.

The problem of hypothesis testing can also be studied in its general formulations: given two (abstract) hypothesis H_0 and H_1 about the unknown measure that generates the data, find out whether it is possible to test H_0 against H_1 (with confidence), and if so, how can one do it.

3.3.3. Change Point Analysis

A stochastic process is generating the data. At some point, the process distribution changes. In the "offline" situation, the statistician observes the resulting sequence of outcomes and has to estimate the point or the points at which the change(s) occurred. In online setting, the goal is to detect the change as quickly as possible.

These are the classical problems in mathematical statistics, and probably among the last remaining statistical problems not adequately addressed by machine learning methods. The reason for the latter is perhaps in that the problem is rather challenging. Thus, most methods available so far are parametric methods concerning piece-wise constant distributions, and the change in distribution is associated with the change in the mean. However, many applications, including DNA analysis, the analysis of (user) behavior data, etc., fail to comply with this kind of assumptions. Thus, our goal here is to provide completely non-parametric methods allowing for any kind of changes in the time-series distribution.

3.3.4. Clustering Time Series, Online and Offline

The problem of clustering, while being a classical problem of mathematical statistics, belongs to the realm of unsupervised learning. For time series, this problem can be formulated as follows: given several samples $x^1 = (x_1^1, \cdots, x_{n_1}^1), \cdots, x^N = (x_N^1, \cdots, x_{n_N}^N)$, we wish to group similar objects together. While this is of course not a precise formulation, it can be made precise if we assume that the samples were generated by k different distributions.

The online version of the problem allows for the number of observed time series to grow with time, in general, in an arbitrary manner.

3.3.5. Online Semi-Supervised Learning

Semi-supervised learning (SSL) is a field of machine learning that studies learning from both labeled and unlabeled examples. This learning paradigm is extremely useful for solving real-world problems, where data is often abundant but the resources to label them are limited.

Furthermore, *online* SSL is suitable for adaptive machine learning systems. In the classification case, learning is viewed as a repeated game against a potentially adversarial nature. At each step t of this game, we observe an example $\mathbf{x_t}$, and then predict its label $\widehat{y}_t$.

The challenge of the game is that we only exceptionally observe the true label y_t. In the extreme case, which we also study, only a handful of labeled examples are provided in advance and set the initial bias of the system while unlabeled examples are gathered online and update the bias continuously. Thus, if we want to adapt to changes in the environment, we have to rely on indirect forms of feedback, such as the structure of data.

3.3.6. Online Kernel and Graph-Based Methods

Large-scale kernel ridge regression is limited by the need to store a large kernel matrix. Similarly, large-scale graph-based learning is limited by storing the graph Laplacian. Furthermore, if the data come online, at some point no finite storage is sufficient and per step operations become slow.

Our challenge is to design sparsification methods that give guaranteed approximate solutions with a reduced storage requirements.

VALSE Project-Team

3. Research Program

3.1. Research Program

Valse team works in the domains of control science: dynamical systems, stability analysis, estimation and automatic control. *Our developments are focused on the theoretical and applied aspects related to control and estimation of large-scale multi-sensor and multi-actuator systems based on the use of the theories of finite-time/fixed-time/hyperexponential convergence and homogeneous systems*. The Lyapunov function method and other methods of analysis of dynamical systems form a basis for the studies in Valse team.

The key idea of research program for the team is that a fast (non-asymptotic) convergence of the regulation and estimation errors increases the reliability of intelligent distributed actuators and sensors in complex scenarios, such as interconnected cyber-physical systems (CPSs).

The expertise of Valse's members in theoretical developments of control and estimation theory (finite-time control and estimation algorithms in centralized context [84], [70], [81], [80], [77], homogeneity framework for differential equations [85], [72], [71], [73], [75], [86], [82], time-delay systems [74], [76], [89], distributed systems [83] and algebraic-based methods for estimation [87], [88]) is an essential ingredient to achieve our objective.

The generic chart of different goals and tasks included in the scientific work program of Valse, and interrelations between them, are presented in Fig. 1 . We have selected three main objectives to pursuit with the related tasks to fulfill:

- The first objective consists in design of control and estimation solutions for CPS and IoT, which is the principal aim of Valse, it will contain the main outcomes of our research.
- The second objective is more theoretical, which is needed to make the basement for our design and analysis parts in the previous goal.
- The third objective deals with applications, which will drive the team and motivate the theoretical studies and selected design performances.

All these objectives are interconnected: from a particular problem in an IoT application, it is planned to design a control or estimation algorithm, which leads to development of theoretical tools; and *vice versa*, a new theoretical advance can provide a possibility for development of novel tools, which can be used in applications.

To explain our motivation: *why to use finite-time?* Applying any method for control/estimation has a price in terms of its advantages and disadvantages. There is no universal framework that is the best always and everywhere. Finite-time may appear as a luxurious property for a physical system, requiring the use of nonlinear tools. Of course, if an asymptotic convergence and a linear model are enough for solving a given problem, then there is no reason to develop something else. However, most of the present problems in CPS and IoT are nonlinear (i.e. they have various local behaviors that cannot be collected in only one linear model). Design and analysis of various local linearized models and solutions are luxurious, too. The theory of homogeneity can go beyond linearity offering many new features, while not appearing as severe as other nonlinear tools and having almost all hints of the linear framework. Suppose that, thanks to the homogeneity theory, finite-time/fixed-time can be obtained with a limited difficulty, while adding the bonuses of a stronger robustness and a faster convergence compared to the linear case? *We are convinced that the price of going beyond linear control and estimation can be strongly dropped down by maturing the theory of homogeneity and finite/fixed-time convergence. And also, convinced that it will be compensated in terms of robustness and speed, which can be demanded in the new areas of application as IoT, for example*.

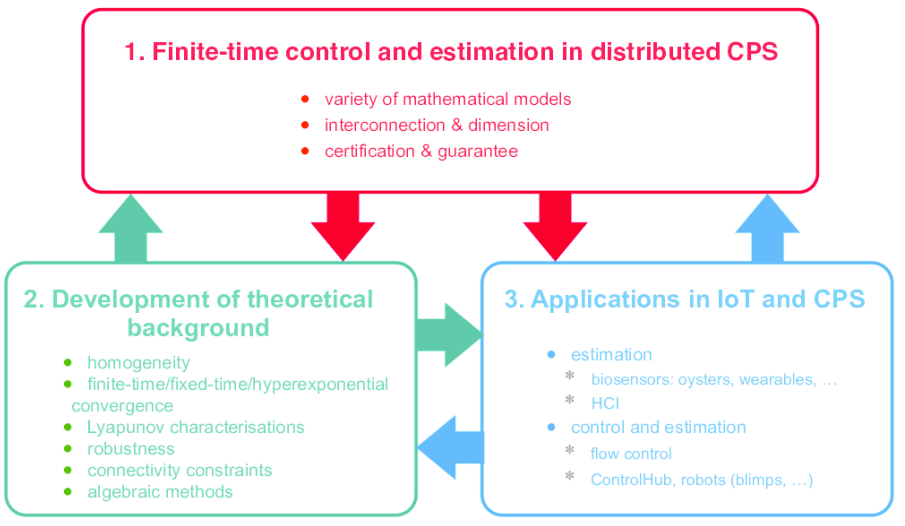

Figure 1. Structure of the objectives and tasks treated in Valse

FUN Project-Team

3. Research Program

3.1. Introduction

We will focus on wireless ubiquitous networks that rely on constrained devices, i.e. with limited resources in terms of storage and computing capacities. They can be sensors, small robots, RFID readers or tags. A wireless sensor retrieves a physical measure such as light. A wireless robot is a wireless sensor that in addition has the ability to move by itself in a controlled way. A drone is a robot with the ability to manoeuvre in 3D (in the air or in the water). RFID tags are passive items that embed a unique identifier for a place or an object allowing accurate traceability. They can communicate only in the vicinity of an RFID reader. An RFID reader can be seen as a special kind of sensor in the network which data is the one read on tags. These devices may run on batteries that are not envisaged to be changed or recharged. These networks may be composed of ten to thousands of such heterogeneous devices for which energy is a key issue.

Today, most of these networks are homogeneous, i.e. composed of only one kind of devices. They have mainly been studied in application and technology silos. Because of this, they are approaching fundamental limitations especially in terms of topology deployment, management and communications, while exploiting the complementarity of heterogeneous devices and communication technologies would enlarge their capacities and the set of applications. Finally, these networks must work efficiently even in dynamic and realistic situations, i.e. they must consider by design the different dynamic parameters and automatically self-adapt to their variations.

Our overall goal is represented by Figure 1 . We will investigate wireless ubiquitous IoT services for constrained devices by smartly combining **different frequency bands** and **different medium access and routing techniques** over **heterogeneous devices** in a **distributed** and **opportunistic** fashion. Our approach will always deal with **hardware constraints** and take care of **security** and **energy** issues to provide protocols that ride on **synergy** and **self-organization** between devices.

The goal of the FUN project team is to provide these next generation networks with a set of innovative and distributed self-organizing cooperative protocols to raise them to a new level of scalability, autonomy, adaptability, manageability and performance. We aim to break these silos to exploit the full synergy between devices, making them cooperate in a single holistic network. We will consider them as networks of heterogeneous devices rather than a collection of heterogeneous networks.

To realize the full potential of these ubiquitous networks, there is a need to provide them with a set of tools that allow them to *(i)* (self-)deploy, *(ii)* self-organize, *(iii)* discover and locate each other, resources and services and *(iv)* communicate. These tools will be the basics for enabling cooperation, co-existence and witnessing a global efficient behavior. The deployment of these mechanisms is challenging since it should be achieved in spite of several limitations. The main difficulties are to provide such protocols in a **secured** and **energy-efficient** fashion in spite of:

- dynamic topology changes due to various factors such as the unreliability of the wireless medium, the wireless interferences between devices, node mobility and energy saving mechanisms;
- hardware constraints in terms of CPU and memory capacities that limit the operations and data each node can perform/collect;
- lacks of interoperability between applicative, hardware and technological silos that may prevent from data exchange between different devices.

3.1.1. Objectives and methodology

To reach our overall goal, we will pursue the two following objectives. These two objectives are orthogonal and can be carried on jointly:

1. Providing realistic complete self-organizing tools *e.g. vertical perspective*.
2. Going to heterogeneous energy-efficient performing wireless networks *e.g. horizontal perspective*.

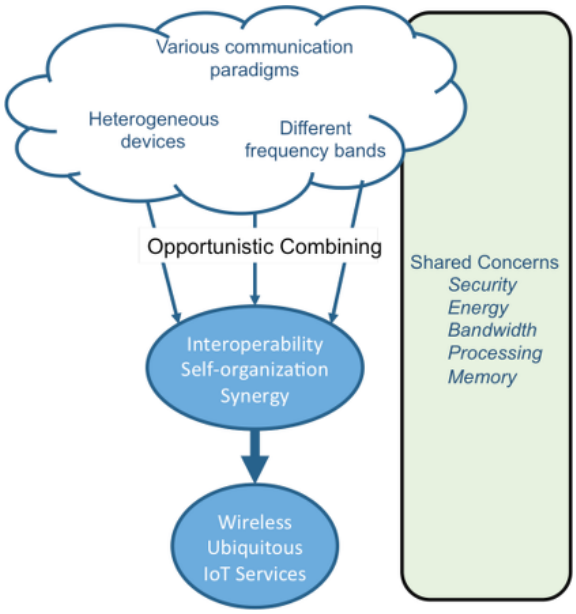

Figure 1. FUN's overal goal.

We give more details on these two objectives below. To achieve our main objectives, we will mainly apply the methodology depicted in Figure 2 combining both theoretical analysis and experimental validation. Mathematical tools will allow us to properly dimension a problem, formally define its limitations and needs to provide suitable protocols in response. Then, they will allow us to qualify the outcome solutions before we validate and stress them in real scenarios with regards to applications requirements. For this, we will realize proofs-of-concept with real scenarios and real devices. Differences between results and expectations will be analyzed in return in order to well understand them and integrate them by design for a better protocol self-adaptation capability.

3.2. Vertical Perspective

As mentioned, future ubiquitous networks evolve in dynamic and unpredictable environments. Also, they can be used in a large scope of applications that have several expectations in terms of performance and different contextual limitations. In this heterogeneous context, IoT devices must support multiple applications and relay traffic with non-deterministic pattern.

To make our solutions practical and efficient in real conditions, we will adopt the dual approach both *top-down* and *bottom-up*. The *top-down* approach will ensure that we consider the application (such as throughput, delay, energy consumption, etc.) and environmental limitations (such as deployment constraints, etc.). The *bottom-up* approach will ensure that we take account of the physical and hardware characteristics such as memory, CPU, energy capacities but also physical interferences and obstacles. With this integrated perpective, we will be in capacity to design well adapted **cross-layer** integrated protocols [59]. We will design jointly routing and MAC layers by taking dynamics occurring at the physical layer into account with a constant concern for energy and security. We will investigate new adaptive frequency hopping techniques combined with routing protocols [59], [45].

This vision will also allow us to integrate external factors by design in our protocols, in an opportunistic way. Yet, we will leverage on the occurrence of any of these phenomena rather than perceiving them as obstacles

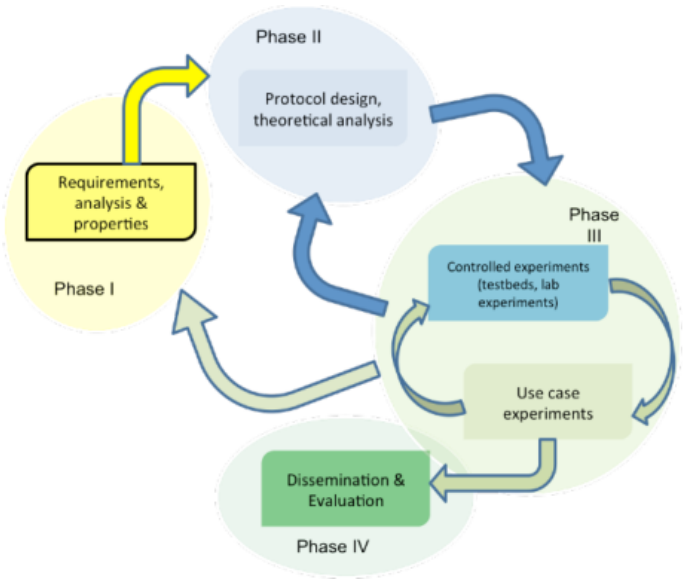

Figure 2. Methodology to be applied in FUN.

or limitations. As an example, we will rely on node undergone mobility to enhance routing performance as we have started to investigate in [55], [37]. On the same idea, when specific features are available like controlled mobility, we will exploit it to improve connectivity or coverage quality like in [49], [57], [31], [25].

3.3. Horizontal perspective

We aim at designing efficient tools for a plethora of wireless devices supporting highly heterogeneous technologies. We will thus investigate these networks from a horizontal perspective, e.g. by considering heterogeneity in low level communications layers.

Given the spectrum scarcity, they will probably need to coexist in the same frequency bands and sometimes for different purposes (RFID tag reading may use the same frequency bands as the wireless sensors). One important aspect to consider in this setting is how these different access technologies will interact with each other, and what are the mechanisms needed to be put in place to guarantee that all services obtain the required share of resources when needed. This problem appears in different application domains, ranging from traffic offloading to unlicensed bands by cellular networks and the need to coexist with WiFi and radars, from a scenario in which multiple-purpose IoT clouds coexist in a city [56]. We will thus explore the dynamics of these interactions and devise ways to ensure smooth coexistence while considering the heterogeneity of the devices involved, the access mechanisms used as well as the requirements of the services provided.

To face the spectrum scarcity, we will also investigate new alternative communication paradigms such as phonon-based or light-based communications as we have initiated in [42] and we will work on the coexistence of these technologies with traditional communication techniques, specifically by investigating efficient switching techniques from one communication technology to the other (they were most focused on the security aspects, to prevent jamming attacks). Resilience and reliability of the whole system will be the key factors to be taken into account [43], [39], [18].

As a more prospective activity, we consider exploring software and communication security for IoT. This is challenging given that existing solutions do not address systems that are both constrained and networked [44].

Finally, in order to contribute to a better interoperability between all these technologies, we will continue to contribute to standardization bodies such as IETF and EPC Global.

RMOD Project-Team

3. Research Program

3.1. Software Reengineering

Strong coupling among the parts of an application severely hampers its evolution. Therefore, it is crucial to answer the following questions: How to support the substitution of certain parts while limiting the impact on others? How to identify reusable parts? How to modularize an object-oriented application?

Having good classes does not imply a good application layering, absence of cycles between packages and reuse of well-identified parts. Which notion of cohesion makes sense in presence of late-binding and programming frameworks? Indeed, frameworks define a context that can be extended by subclassing or composition: in this case, packages can have a low cohesion without being a problem for evolution. How to obtain algorithms that can be used on real cases? Which criteria should be selected for a given remodularization?

To help us answer these questions, we work on enriching Moose, our reengineering environment, with a new set of analyses [31], [30]. We decompose our approach in three main and potentially overlapping steps:

1. Tools for understanding applications,
2. Remodularization analyses,
3. Software Quality.

3.1.1. Tools for understanding applications

Context and Problems. We are studying the problems raised by the understanding of applications at a larger level of granularity such as packages or modules. We want to develop a set of conceptual tools to support this understanding.

Some approaches based on Formal Concept Analysis (FCA) [59] show that such an analysis can be used to identify modules. However the presented examples are too small and not representative of real code.

Research Agenda.

FCA provides an important approach in software reengineering for software understanding, design anomalies detection and correction, but it suffers from two problems: (i) it produces lattices that must be interpreted by the user according to his/her understanding of the technique and different elements of the graph; and, (ii) the lattice can rapidly become so big that one is overwhelmed by the mass of information and possibilities [20]. We look for solutions to help people putting FCA to real use.

3.1.2. Remodularization analyses

Context and Problems. It is a well-known practice to layer applications with bottom layers being more stable than top layers [47]. Until now, few works have attempted to identify layers in practice: Mudpie [61] is a first cut at identifying cycles between packages as well as package groups potentially representing layers. DSM (dependency structure matrix) [60], [55] seems to be adapted for such a task but there is no serious empirical experience that validates this claim. From the side of remodularization algorithms, many were defined for procedural languages [43]. However, object-oriented programming languages bring some specific problems linked with late-binding and the fact that a package does not have to be systematically cohesive since it can be an extension of another one [62], [34].

As we are designing and evaluating algorithms and analyses to remodularize applications, we also need a way to understand and assess the results we are obtaining.

Research Agenda. We work on the following items:

Layer identification. We propose an approach to identify layers based on a semi-automatic classification of package and class interrelationships that they contain. However, taking into account the wish or knowledge of the designer or maintainer should be supported.

Cohesion Metric Assessment. We are building a validation framework for cohesion/coupling metrics to determine whether they actually measure what they promise to. We are also compiling a number of traditional metrics for cohesion and coupling quality metrics to evaluate their relevance in a software quality setting.

3.1.3. Software Quality

Research Agenda. Since software quality is fuzzy by definition and a lot of parameters should be taken into account we consider that defining precisely a unique notion of software quality is definitively a Grail in the realm of software engineering. The question is still relevant and important. We work on the two following items:

Quality models. We studied existing quality models and the different options to combine indicators — often, software quality models happily combine metrics, but at the price of losing the explicit relationships between the indicator contributions. There is a need to combine the results of one metric over all the software components of a system, and there is also the need to combine different metric results for any software component. Different combination methods are possible that can give very different results. It is therefore important to understand the characteristics of each method.

Bug prevention. Another aspect of software quality is validating or monitoring the source code to avoid the emergence of well known sources of errors and bugs. We work on how to best identify such common errors, by trying to identify earlier markers of possible errors, or by helping identifying common errors that programmers did in the past.

3.2. Language Constructs for Modular Design

While the previous axis focuses on how to help remodularizing existing software, this second research axis aims at providing new language constructs to build more flexible and recomposable software. We will build on our work on traits [57], [32] and classboxes [21] but also start to work on new areas such as isolation in dynamic languages. We will work on the following points: (1) Traits and (2) Modularization as a support for isolation.

3.2.1. Traits-based program reuse

Context and Problems. Inheritance is well-known and accepted as a mechanism for reuse in object-oriented languages. Unfortunately, due to the coarse granularity of inheritance, it may be difficult to decompose an application into an optimal class hierarchy that maximizes software reuse. Existing schemes based on single inheritance, multiple inheritance, or mixins, all pose numerous problems for reuse.

To overcome these problems, we designed a new composition mechanism called Traits [57], [32]. Traits are pure units of behavior that can be composed to form classes or other traits. The trait composition mechanism is an alternative to multiple or mixin inheritance in which the composer has full control over the trait composition. The result enables more reuse than single inheritance without introducing the drawbacks of multiple or mixin inheritance. Several extensions of the model have been proposed [29], [51], [22], [33] and several type systems were defined [35], [58], [52], [45].

Traits are reusable building blocks that can be explicitly composed to share methods across unrelated class hierarchies. In their original form, traits do not contain state and cannot express visibility control for methods. Two extensions, stateful traits and freezable traits, have been proposed to overcome these limitations. However, these extensions are complex both to use for software developers and to implement for language designers.

Research Agenda: Towards a pure trait language. We plan distinct actions: (1) a large application of traits, (2) assessment of the existing trait models and (3) bootstrapping a pure trait language.

- To evaluate the expressiveness of traits, some hierarchies were refactored, showing code reuse [24]. However, such large refactorings, while valuable, may not exhibit all possible composition problems, since the hierarchies were previously expressed using single inheritance and following certain patterns. We want to redesign from scratch the collection library of Smalltalk (or part of it). Such a redesign should on the one hand demonstrate the added value of traits on a real large and redesigned library and on the other hand foster new ideas for the bootstrapping of a pure trait-based language.

  In particular we want to reconsider the different models proposed (stateless [32], stateful [23], and freezable [33]) and their operators. We will compare these models by (1) implementing a trait-based collection hierarchy, (2) analyzing several existing applications that exhibit the need for traits. Traits may be flattened [50]. This is a fundamental property that confers to traits their simplicity and expressiveness over Eiffel's multiple inheritance. Keeping these aspects is one of our priority in forthcoming enhancements of traits.
- Alternative trait models. This work revisits the problem of adding state and visibility control to traits. Rather than extending the original trait model with additional operations, we use a fundamentally different approach by allowing traits to be lexically nested within other modules. This enables traits to express (shared) state and visibility control by hiding variables or methods in their lexical scope. Although the traits' "flattening property" no longer holds when they can be lexically nested, the combination of traits with lexical nesting results in a simple and more expressive trait model. We formally specify the operational semantics of this combination. Lexically nested traits are fully implemented in AmbientTalk, where they are used among others in the development of a Morphic-like UI framework.
- We want to evaluate how inheritance can be replaced by traits to form a new object model. For this purpose we will design a minimal reflective kernel, inspired first from ObjVlisp [28] then from Smalltalk [38].

3.2.2. Reconciling Dynamic Languages and Isolation

Context and Problems. More and more applications require dynamic behavior such as modification of their own execution (often implemented using reflective features [42]). For example, F-script allows one to script Cocoa Mac-OS X applications and Lua is used in Adobe Photoshop. Now in addition more and more applications are updated on the fly, potentially loading untrusted or broken code, which may be problematic for the system if the application is not properly isolated. Bytecode checking and static code analysis are used to enable isolation, but such approaches do not really work in presence of dynamic languages and reflective features. Therefore there is a tension between the need for flexibility and isolation.

Research Agenda: Isolation in dynamic and reflective languages. To solve this tension, we will work on *Sure*, a language where isolation is provided by construction: as an example, if the language does not offer field access and its reflective facilities are controlled, then the possibility to access and modify private data is controlled. In this context, layering and modularizing the meta-level [25], as well as controlling the access to reflective features [26], [27] are important challenges. We plan to:

- Study the isolation abstractions available in erights (http://www.erights.org) [49], [48], and Java's class loader strategies [44], [39].
- Categorize the different reflective features of languages such as CLOS [41], Python and Smalltalk [53] and identify suitable isolation mechanisms and infrastructure [36].
- Assess different isolation models (access rights, capabilities [54],...) and identify the ones adapted to our context as well as different access and right propagation.
- Define a language based on
  - the decomposition and restructuring of the reflective features [25],

- the use of encapsulation policies as a basis to restrict the interfaces of the controlled objects [56],
- the definition of method modifiers to support controlling encapsulation in the context of dynamic languages.

An open question is whether, instead of providing restricted interfaces, we could use traits to grant additional behavior to specific instances: without trait application, the instances would only exhibit default public behavior, but with additional traits applied, the instances would get extra behavior. We will develop *Sure*, a modular extension of the reflective kernel of Smalltalk (since it is one of the languages offering the largest set of reflective features such as pointer swapping, class changing, class definition,...) [53].

SPIRALS Project-Team

3. Research Program

3.1. Introduction

Our research program on self-adaptive software targets two key properties that are detailed in the remainder of this section: *self-healing* and *self-optimization*.

3.2. Objective #1: Self-healing - Mining software artifacts to automatically evolve systems

Software systems are under the pressure of changes all along their lifecycle. Agile development blurs the frontier between design and execution and requires constant adaptation. The size of systems (millions of lines of code) multiplies the number of bugs by the same order of magnitude. More and more systems, such as sensor network devices, live in "surviving" mode, in the sense that they are neither rebootable nor upgradable.

Software bugs are hidden in source code and show up at development-time, testing-time or worse, once deployed in production. Except for very specific application domains where formal proofs are achievable, bugs can not be eradicated. As an order of magnitude, on 16 Dec 2011, the Eclipse bug repository contains 366 922 bug reports. Software engineers and developers work on bug fixing on a daily basis. Not all developers spend the same time on bug fixing. In large companies, this is sometimes a full-time role to manage bugs, often referred to as *Quality Assurance* (QA) software engineers. Also, not all bugs are equal, some bugs are analyzed and fixed within minutes, others may take months to be solved [79].

In terms of research, this means that: (i) one needs means to automatically adapt the design of the software system through automated refactoring and API extraction, (ii) one needs approaches to automate the process of adapting source code in order to fix certain bugs, (iii) one needs to revisit the notion of error-handling so that instead of crashing in presence of errors, software adapts itself to continue with its execution, *e.g.*, in degraded mode.

There is no one-size-fits-all solution for each of these points. However, we think that novel solutions can be found by using **data mining and machine learning techniques tailored for software engineering** [80]. This body of research consists of mining some knowledge about a software system by analyzing the source code, the version control systems, the execution traces, documentation and all kinds of software development and execution artifacts in general. This knowledge is then used within recommendation systems for software development, auditing tools, runtime monitors, frameworks for resilient computing, etc.

The novelty of our approach consists of using and tailoring data mining techniques for analyzing software artifacts (source code, execution traces) in order to achieve the **next level of automated adaptation** (*e.g.*, automated bug fixing). Technically, we plan to mix unsupervised statistical learning techniques (*e.g.* frequent item set mining) and supervised ones (*e.g.* training classifiers such as decision trees). This research is currently not being performed by data mining research teams since it requires a high level of domain expertise in software engineering, while software engineering researchers can use off-the-shelf data mining libraries, such as Weka [58].

We now detail the two directions that we propose to follow to achieve this objective.

3.2.1. Learning from software history how to design software and fix bugs

The first direction is about mining techniques in software repositories (*e.g.*, CVS, SVN, Git). Best practices can be extracted by data mining source code and the version control history of existing software systems. The design and code of expert developers significantly vary from the artifacts of novice developers. We will learn to differentiate those design characteristics by comparing different code bases, and by observing the semantic refactoring actions from version control history. Those design rules can then feed the test-develop-refactor constant adaptation cycle of agile development.

Fault localization of bugs reported in bug repositories. We will build a solid foundation on empirical knowledge about bugs reported in bug repository. We will perform an empirical study on a set of representative bug repositories to identify classes of bugs and patterns of bug data. For this, we will build a tool to browse and annotate bug reports. Browsing will be helped with two kinds of indexing: first, the tool will index all textual artifacts for each bug report; second it will index the semantic information that is not present by default in bug management software—*i.e.*, "contains a stacktrace"). Both indexes will be used to find particular subsets of bug reports, for instance "all bugs mentioning invariants and containing a stacktrace". Note that queries with this kind of complexity and higher are mostly not possible with the state-of-the-art of bug management software. Then, analysts will use annotation features to annotate bug reports. The main outcome of the empirical study will be the identification of classes of bugs that are appropriate for automated localization. Then, we will run machine learning algorithms to identify the latent links between the bug report content and source code features. Those algorithms would use as training data the existing traceability links between bug reports and source code modifications from version control systems. We will start by using decision trees since they produce a model that is explicit and understandable by expert developers. Depending on the results, other machine learning algorithms will be used. The resulting system will be able to locate elements in source code related to a certain bug report with a certain confidence.

Automated bug fix generation with search-based techniques. Once a location in code is identified as being the cause of the bug, we can try to automatically find a potential fix. We envision different techniques: (1) infer fixes from existing contracts and specifications that are violated; (2) infer fixes from the software behavior specified as a test suite; (3) try different fix types one-by-one from a list of identified bug fix patterns; (4) search fixes in a fix space that consists of combinations of atomic bug fixes. Techniques 1 and 2 are explored in [54] and [78]. We will focus on the latter techniques. To identify bug fix patterns and atomic bug fixes, we will perform a large-scale empirical study on software changes (also known as changesets when referring to changes across multiple files). We will develop tools to navigate, query and annotate changesets in a version control system. Then, a grounded theory will be built to master the nature of fixes. Eventually, we will decompose change sets in atomic actions using clustering on changeset actions. We will then use this body of empirical knowledge to feed search-based algorithms (*e.g.* genetic algorithms) that will look for meaningful fixes in a large fix space. To sum up, our research on automated bug fixing will try not only to point to source code locations responsible of a bug, but to search for code patterns and snippets that may constitute the skeleton of a valid patch. Ultimately, a blend of expert heuristics and learned rules will be able to produce valid source code that can be validated by developers and committed to the code base.

3.2.2. Run-time self-healing

The second proposed research direction is about inventing a self-healing capability at run-time. This is complementary to the previous objective that mainly deals with development time issues. We will achieve this in two steps. First, we want to define frameworks for resilient software systems. Those frameworks will help to maintain the execution even in the presence of bugs—*i.e.* to let the system survive. As exposed below, this may mean for example to switch to some degraded modes. Next, we want to go a step further and to define solutions for automated runtime repair, that is, not simply compensating the erroneous behavior, but also determining the correct repair actions and applying them at run-time.

Mining best effort values. A well-known principle of software engineering is the "fail-fast" principle. In a nutshell, it states that as soon as something goes wrong, software should stop the execution before entering incorrect states. This is fine when a human user is in the loop, capable of understanding the error or at least rebooting the system. However, the notion of "failure-oblivious computing" [71] shows that in certain domains, software should run in a resilient mode (*i.e.* capable of recovering from errors) and/or best-effort mode—*i.e.* a slightly imprecise computation is better than stopping. Hence, we plan to investigate data mining techniques in order to learn best-effort values from past executions (*i.e.* somehow learning what is a correct state, or the opposite what is not a completely incorrect state). This knowledge will then be used to adapt the software state and flow in order to mitigate the error consequences, the exact opposite of fail-fast for systems with long-running cycles.

Embedding search based algorithms at runtime. Harman recently described the field of search-based software engineering [59]. We believe that certain search based approaches can be embedded at runtime with the goal of automatically finding solutions that avoid crashing. We will create software infrastructures that allow automatically detecting and repairing faults at run-time. The methodology for achieving this task is based on three points: (1) empirical study of runtime faults; (2) learning approaches to characterize runtime faults; (3) learning algorithms to produce valid changes to the software runtime state. An empirical study will be performed to analyze those bug reports that are associated with runtime information (*e.g.* core dumps or stacktraces). After this empirical study, we will create a system that learns on previous repairs how to produce small changes that solve standard runtime bugs (*e.g.* adding an array bound check to throw a handled domain exception rather than a spurious language exception). To achieve this task, component models will be used to (1) encapsulate the monitoring and reparation meta-programs in appropriate components and (2) support runtime code modification using scripting, reflective or bytecode generation techniques.

3.3. Objective #2: Self-optimization - Sharing runtime behaviors to continuously adapt software

Complex distributed systems have to seamlessly adapt to a wide variety of deployment targets. This is due to the fact that developers cannot anticipate all the runtime conditions under which these systems are immersed. A major challenge for these software systems is to develop their capability to continuously reason about themselves and to take appropriate decisions and actions on the optimizations they can apply to improve themselves. This challenge encompasses research contributions in different areas, from environmental monitoring to real-time symptoms diagnosis, to automated decision making. The variety of distributed systems, the number of optimization parameters, and the complexity of decisions often resign the practitioners to design monolithic and static middleware solutions. However, it is now globally acknowledged that the development of dedicated building blocks does not contribute to the adoption of sustainable solutions. This is confirmed by the scale of actual distributed systems, which can—for example—connect several thousands of devices to a set of services hosted in the Cloud. In such a context, the lack of support for smart behaviors at different levels of the systems can inevitably lead to its instability or its unavailability. In June 2012, an outage of Amazon's Elastic Compute Cloud in North Virginia has taken down Netflix, Pinterest, and Instagram services. During hours, all these services failed to satisfy their millions of customers due to the lack of integration of a self-optimization mechanism going beyond the boundaries of Amazon.

The research contributions we envision within this area will therefore be organized as a reference model for engineering **self-optimized distributed systems** autonomously driven by *adaptive feedback control loops*, which will automatically enlarge their scope to cope with the complexity of the decisions to be taken. This solution introduces a multi-scale approach, which first privileges local and fast decisions to ensure the homeostasis [0] property of a single node, and then progressively propagates symptoms in the network in order to reason on a longer term and a larger number of nodes. Ultimately, domain experts and software developers can be automatically involved in the decision process if the system fails to find a satisfying solution. The research program for this objective will therefore focus on the study of mechanisms for **monitoring, taking decisions, and automatically reconfiguring software at runtime and at various scales**. As stated in the self-healing objective, we believe that there is no one-size-fits-all mechanism that can span all the scales of the system. We will therefore study and identify an optimal composition of various adaptation mechanisms in order to produce long-living software systems.

The novelty of this objective is to exploit the wisdom of crowds to define new middleware solutions that are able to continuously adapt software deployed in the wild. We intend to demonstrate the applicability of this approach to distributed systems that are deployed from mobile phones to cloud infrastructures. The key scientific challenges to address can be summarized as follows: *How does software behave once deployed in the wild? Is it possible to automatically infer the quality of experience, as it is perceived by users? Can the*

[0] Homeostasis is the property of a system that regulates its internal environment and tends to maintain a stable, relatively constant condition of properties [Wikipedia].

runtime optimizations be shared across a wide variety of software? How optimizations can be safely operated on large populations of software instances?

The remainder of this section further elaborates on the opportunities that can be considered within the frame of this objective.

3.3.1. Monitoring software in the wild

Once deployed, developers are generally no longer aware of how their software behave. Even if they heavily use testbeds and benchmarks during the development phase, they mostly rely on the bugs explicitly reported by users to monitor the efficiency of their applications. However, it has been shown that contextual artifacts collected at runtime can help to understand performance leaks and optimize the resilience of software systems [81]. Monitoring and understanding the context of software at runtime therefore represent the first building block of this research challenge. Practically, we intend to investigate crowd-sensing approaches, to smartly collect and process runtime metrics (*e.g.*, request throughput, energy consumption, user context). Crowd-sensing can be seen as a specific kind of crowdsourcing activity, which refers to the capability of lifting a (large) diffuse group of participants to delegate the task of retrieving trustable data from the field. In particular, crowd-sensing covers not only *participatory sensing* to involve the user in the sensing task (*e.g.*, surveys), but also *opportunistic sensing* to exploit mobile sensors carried by the user (*e.g.*, smartphones).

While reported metrics generally enclose raw data, the monitoring layer intends to produce meaningful indicators like the *Quality of Experience* (QoE) perceived by users. This QoE reflects representative symptoms of software requiring to trigger appropriate decisions in order to improve its efficiency. To diagnose these symptoms, the system has to process a huge variety of data including runtime metrics, but also history of logs to explore the sources of the reported problems and identify opportunities for optimizations. The techniques we envision at this level encompass machine learning, principal component analysis, and fuzzy logic [70] to provide enriched information to the decision level.

3.3.2. Collaborative decision-making approaches

Beyond the symptoms analysis, decisions should be taken in order to improve the *Quality of Service* (QoS). In our opinion, collaborative approaches represent a promising solution to effectively converge towards the most appropriate optimization to apply for a given symptom. In particular, we believe that exploiting the wisdom of the crowd can help the software to optimize itself by sharing its experience with other software instances exhibiting similar symptoms. The intuition here is that the body of knowledge that supports the optimization process cannot be specific to a single software instance as this would restrain the opportunities for improving the quality and the performance of applications. Rather, we think that any software instance can learn from the experience of others.

With regard to the state-of-the-art, we believe that a multi-levels decision infrastructure, inspired from distributed systems like Spotify [57], can be used to build a decentralized decision-making algorithm involving the surrounding peers before requesting a decision to be taken by more central control entity. In the context of collaborative decision-making, peer-based approaches therefore consist in quickly reaching a consensus on the decision to be adopted by a majority of software instances. Software instances can share their knowledge through a micro-economic model [51], that would weight the recommendations of experienced instances, assuming their age reflects an optimal configuration.

Beyond the peer level, the adoption of algorithms inspired from evolutionary computations, such as genetic programming, at an upper level of decision can offer an opportunity to test and compare several alternative decisions for a given symptom and to observe how does the crowd of applications evolves. By introducing some diversity within this population of applications, some instances will not only provide a satisfying QoS, but will also become naturally resilient to unforeseen situations.

3.3.3. Smart reconfigurations in the large

Any decision taken by the crowd requires to propagate back to and then operated by the software instances. While simplest decisions tend to impact software instances located on a single host (*e.g.*, laptop, smartphone),

this process can also exhibit more complex reconfiguration scenarios that require the orchestration of various actions that have to be safely coordinated across a large number of hosts. While it is generally acknowledged that centralized approaches raise scalability issues, we think that self-optimization should investigate different reconfiguration strategies to propagate and apply the appropriate actions. The investigation of such strategies can be addressed in two steps: the consideration of *scalable data propagation protocols* and the identification of *smart reconfiguration mechanisms*.

With regard to the challenge of scalable data propagation protocols, we think that research opportunities encompass not only the exploitation of gossip-based protocols [56], but also the adoption of publish/subscribe abstractions [64] in order to decouple the decision process from the reconfiguration. The fundamental issue here is the definition of a communication substrate that can accommodate the propagation of decisions with relaxed properties, inspired by *Delay Tolerant Networks* (DTN), in order to reach weakly connected software instances. We believe that the adoption of asynchronous communication protocols can provide the sustainable foundations for addressing various execution environments including harsh environments, such as developing countries, which suffer from a partial connectivity to the network. Additionally, we are interested in developing the principle of *social networks of applications* in order to seamlessly group and organize software instances according to their similarities and acquaintances. The underlying idea is that grouping application instances can contribute to the identification of optimization profiles not only contributing to the monitoring layer, but also interested in similar reconfigurations. Social networks of applications can contribute to the anticipation of reconfigurations by exploiting the symptoms of similar applications to improve the performance of others before that problems actually happen.

With regard to the challenge of smart reconfiguration mechanisms, we are interested in building on our established experience of adaptive middleware [75] in order to investigate novel approaches to efficient application reconfigurations. In particular, we are interested in adopting seamless micro-updates and micro-reboot techniques to provide in-situ reconfiguration of pieces of software. Additionally, the provision of safe and secured reconfiguration mechanisms is clearly a key issue that requires to be carefully addressed in order to avoid malicious exploitation of dynamic reconfiguration mechanisms against the software itself. In this area, although some reconfiguration mechanisms integrate transaction models [65], most of them are restricted to local reconfigurations, without providing any support for executing distributed reconfiguration transactions. Additionally, none of the approached published in the literature include security mechanisms to preserve from unauthorized or malicious reconfigurations.

DEFROST Project-Team

3. Research Program

3.1. Introduction

Our research crosses different disciplines: numerical mechanics, control design, robotics, optimisation methods and clinical applications. Our organisation aims at facilitating the team work and cross-fertilisation of research results in the group. We have three objectives (1, 2 and 3) that correspond to the main scientific challenges. In addition, we have two transverse objectives that are also highly challenging: the development of a high performance software support for the project (objective 4) and the validation tools and protocols for the models and methods (objective 5).

3.2. Objective 1: Accurate model of soft robot deformation computed in finite time

The objective is to find concrete numerical solutions to the challenge of modelling soft robots with strong real-time constraints. To solve continuum mechanics equations, we will start our research with real-time FEM or equivalent methods that were developed for soft-tissue simulation. We will extend the functionalities to account for the needs of a soft-robotic system:

- Coupling with other physical phenomenons that govern the activity of sensors and actuators (hydraulic, pneumatic, electro-active polymers, shape-memory alloys...).
- Fulfilling the new computational time constraints (harder than surgical simulation for training) and find better tradeoff between cost and precision of numerical solvers using reduced-order modelling techniques with error control.
- Exploring interactive and semi-automatic optimisation methods for design based on obtained solution for fast computation on soft robot models.

3.3. Objective 2: Model based control of soft robot behavior

The focus of this objective is on obtaining a generic methodology for soft robot feedback control. Several steps are needed to design a model based control from FEM approach:

- The fundamental question of the kinematic link between actuators, sensors, effectors and contacts using the most reduced mathematical space must be carefully addressed. We need to find efficient algorithms for real-time projection of non-linear FEM models in order to pose the control problem using the only relevant parameters of the motion control.
- Intuitive remote control is obtained when the user directly controls the effector motion. To add this functionality, we need to obtain real-time inverse models of the soft robots by optimisation. Several criteria will be combined in this optimisation: effector motion control, structural stiffness of the robot, reduce intensity of the contact with the environment...
- Investigating closed-loop approaches using sensor feedback: as sensors cannot monitor all points of the deformable structure, the information provided will only be partial. We will need additional algorithms based on the FEM model to obtain the best possible treatment of the information. The final objective of these models and algorithms is to have robust and efficient feedback control strategies for soft robots. One of the main challenge here is to ensure / prove stability in closed-loop.

3.4. Objective 3: Modeling the interaction with a complex environment

Even if the inherent mechanical compliance of soft robots makes them safer, more robust and particularly adapted to interaction with fragile environments, the contact forces need to be controlled by:

- Setting up real-time modelling and the control methods needed to pilot the forces that the robot imposes on its environment and to control the robot deformations imposed by its environment. Note that if an operative task requires to apply forces on the surrounding structures, the robot must be anchored to other structures or structurally rigidified.
- Providing mechanics models of the environment that include the uncertainties on the geometry and on the mechanical properties, and are capable of being readjusted in real-time.
- Using the visual feedback of the robot behavior to adapt dynamically the models. The observation provided in the image coupled with an inverse accurate model of the robot could transform the soft robot into sensor: as the robot deforms with the contact of the surroundings, we could retrieve some missing parameters of the environment by a smart monitoring of the robot deformations.

3.5. Objective 4: Soft Robotics Software

Expected research results of this project are numerical methods and algorithms that require high-performance computing and suitability with robotic applications. There is no existing software support for such development. We propose to develop our own software, in a suite split into three applications:

- The first one will facilitate the design of deformable robots by an easy passage from CAD software (for the design of the robot) to the FEM based simulation.
- The second one is an anticipative clinical simulator. The aim is to co-design the robotic assistance with the physicians, thanks to a realistic simulation of the procedure or the robotic assistance. This will facilitate the work of reflection on new clinical approaches prior any manufacturing.
- The third one is the control design software. It will provide the real-time solutions for soft robot control developed in the project.

3.6. Objective 5: Validation and application demonstrations

The implementation of experimental validation is a key challenge for the project. On one side, we need to validate the model and control algorithms using concrete test case example in order to improve the modelling and to demonstrate the concrete feasibility of our methods. On the other side, concrete applications will also feed the reflexions on the objectives of the scientific program.

We will build our own experimental soft robots for the validation of objectives 2 and 3 when there is no existing “turn-key” solution. Designing and making our own soft robots, even if only for validation, will help the setting-up of adequate models.

For the validation of objective 4, we will develop “anatomical soft robot”: soft robot with the shape of organs, equipped with sensors (to measure the contact forces) and actuators (to be able to stiffen the walls and recreate natural motion of soft-tissues). We will progressively increase the level of realism of this novel validation set-up to come closer to the anatomical properties.

LINKS Project-Team

3. Research Program

3.1. Background

The main objective of LINKS is to develop methods for querying and managing linked data collections. Even though open linked data is the most prominent example, we will focus on hybrid linked data collections, which are collections of semi-structured datasets in hybrid formats: graph-based, RDF, relational, and NoSQL. The elements of these datasets may be linked, either by pointers or by additional relations between the elements of the different datasets, for instance the "same-as" or "member-of" relations as in RDF.

The advantage of traditional data models is that there exist powerful querying methods and technologies that one might want to preserve. In particular, they come with powerful schemas that constraint the possible manners in which knowledge is represented to a finite number of patterns. The exhaustiveness of these patterns is essential for writing of queries that cover all possible cases. Pattern violations are excluded by schema validation. In contrast, RDF schema languages such as RDFS can only enrich the relations of a dataset by new relations, which also helps for query writing, but which cannot constraint the number of possible patterns, so that they do not come with any reasonable notion of schema validation.

The main weakness of traditional formats, however, is that they do not scale to large data collections as stored on the Web, while the RDF data models scales well to very big collections such as linked open data. Therefore, our objective is to study mixed data collections, some of which may be in RDF format, in which we can lift the advantages of smaller datasets in traditional formats to much larger linked data collections. Such data collections are typically distributed over the internet, where data sources may have rigid query facilities that cannot be easily adapted or extended.

The main assumption that we impose in order to enable the logical approach, is that the given linked data collection must be correct in most dimensions. This means that all datasets are well-formed with respect to their available constraints and schemas, and clean with respect to the data values in most of the components of the relations in the datasets. One of the challenges is to integrate good quality RDF datasets into this setting, another is to clean the incorrect data in those dimensions that are less proper. It remains to be investigated in how far these assumptions can be maintained in realistic applications, and how much they can be weakened otherwise.

For querying linked data collections, the main problems are to resolve the heterogeneity of data formats and schemas, to understand the efficiency and expressiveness of recursive queries, that can follow links repeatedly, to answer queries under constraints, and to optimize query answering algorithms based on static analysis. When linked data is dynamically created, exchanged, or updated, the problems are how to process linked data incrementally, and how to manage linked data collections that change dynamically. In any case (static and dynamic) one needs to find appropriate schema mappings for linking semi-structured datasets. We will study how to automatize parts of this search process by developing symbolic machine learning techniques for linked data collections.

3.2. Querying Heterogeneous Linked Data

Our main objective is to query collections of linked datasets. In the static setting, we consider two kinds of links: explicit links between elements of the datasets, such as equalities or pointers, and logical links between relations of different datasets such as schema mappings. In the dynamic setting, we permit a third kind of links that point to "intentional" relations computable from a description, such as the application of a Web service or the application of a schema mapping.

We believe that collections of linked datasets are usually too big to ensure a global knowledge of all datasets. Therefore, schema mappings and constraints should remain between pairs of datasets. Our main goal is to be able to pose a query on a collection of datasets, while accounting for the possible recursive effects of schema mappings. For illustration, consider a ring of datasets D_1, D_2, D_3 linked by schema mappings M_1, M_2, M_3 that tell us how to complete a database D_i by new elements from the next database in the cycle.

The mappings M_i induce three intentional datasets I_1, I_2, and I_3, such that I_i contains all elements from D_i and all elements implied by M_i from the next intentional dataset in the ring:

$$I_1 = D_1 \cup M_1(I_2), \quad I_2 = D_2 \cup M_2(I_3), \quad I_3 = D_3 \cup M_3(I_1)$$

Clearly, the global information collected by the intentional datasets depends recursively on all three original datasets D_i. Queries to the global information can now be specified as standard queries to the intentional databases I_i. However, we will never materialize the intentional databases I_i. Instead, we can rewrite queries on one of the intentional datasets I_i to recursive queries on the union of the original datasets D_1, D_2, and D_3 with their links and relations. Therefore, a query answering algorithm is needed for recursive queries, that chases the "links" between the D_i in order to compute the part of I_i needed for the purpose of query answering.

This illustrates that we must account for the graph data models when dealing with linked data collections whose elements are linked, and that query languages for such graphs must provide recursion in order to chase links. Therefore, we will have to study graph databases with recursive queries, such as RDF graphs with SPARQL queries, but also other classes of graph databases and queries.

We study schemas and mappings between datasets with different kinds of data models and the complexity of evaluating recursive queries over graphs. In order to use schema mapping for efficiently querying the different datasets, we need to optimize the queries by taking into account the mappings. Therefore, we will study static analysis of schema mappings and recursive queries. Finally, we develop concrete applications in which our fundamental techniques can be applied.

3.3. Managing Dynamic Linked Data

With the quick growth of the information technology on the Web, more and more Web data gets created dynamically every day, for instance by smartphones, industrial machines, users of social networks, and all kinds of sensors. Therefore, large amounts of dynamic data need to be exchanged and managed by various data-centric web services, such as online shops, online newspapers, and social networks.

Dynamic data is often created by the application of some kind of service on the Web. This kind of data is intentional in the same spirit as the intentional data specified by the application of a schema mapping, or the application of some query to the hidden Web. Therefore, we will consider a third kind of links in the dynamic setting, that map to intentional data specified by whatever kind of function application. Such a function can be defined in data-centric programming languages, in the style of Active XML, XSLT, and NoSQL languages.

The dynamicity of data adds a further dimension to the challenges for linked data collections that we described before, while all the difficulties remain valid. One of the new aspects is that intentional data may be produced incrementally, as for instance when exchanged over data streams. Therefore, one needs incremental algorithms able to evaluate queries on incomplete linked data collections, that are extended or updated incrementally. Note that incremental data may be produced without end, such as a Twitter stream, so that one cannot wait for its completion. Instead, one needs to query and manage dynamic data with as low latency as possible. Furthermore, all static analysis problems are to be re-investigated in the presence of dynamic data.

Another aspect of dynamic data is distribution over the Web, and thus parallel processing as in the cloud. This raises the typical problems coming with data distribution: huge data sources cannot be moved without very high costs, while data must be replicated for providing efficient parallel access. This makes it difficult, if not impossible, to update replicated data consistently. Therefore, the consistency assumption has been removed by NoSQL databases for instance, while parallel algorithmic is limited to naive parallelization (i.e. map/reduce) where only few data needs to be exchanged.

We will investigate incremental query evaluation for distributed data-centered programming languages for linked data collections, dynamic updates as needed for linked data management, and static analysis for linked data workflows.

3.4. Linking Graphs

When datasets from independent sources are not linked with existing schema mappings, we would like to investigate symbolic machine learning solutions for inferring such mappings in order to define meaningful links between data from separate sources. This problem can be studied for various kinds of linked data collections. Before presenting the precise objectives, we will illustrate our approach on the example of linking data in two independent graphs: an address book of a research institute containing detailed personnel information and a (global) bibliographic database containing information on papers and their authors.

We remind that a schema allows to identify a collection of types each grouping objects from the same semantic class e.g., the collection of all persons in the address book and the collection of all authors in the bibliography database. As a schema is often lacking or underspecified in graph data models, we intend to investigate inference methods based on structural similarity of graph fragments used to describe objects from the same class in a given document e.g., in the bibliographic database every author has a name and a number of affiliations, while a paper has a title and a number of authors. Furthermore, our inference methods will attempt to identify, for every type, a set of possible keys, where by key we understand a collection of attributes of an object that uniquely identifies such an object in its semantic class. For instance, for a person in the address book two examples of a key are the name of the person and the office phone number of that person.

In the next step, we plan to investigate employing existing entity linkage solutions to identify pairs of types from different databases whose instances should be linked using compatible keys. For instance, persons in the address book should be linked with authors in the bibliographical database using the name as the compatible key. Linking the same objects (represented in different ways) in two databases can be viewed as an instance of a mapping between the two databases. Such mapping is, however, discriminatory because it typically maps objects from a specific subset of objects of given types. For instance, the mapping implied by linking persons in the address book with authors in the bibliographic database involves in fact researchers, a subgroup of personnel of the research institute, and authors affiliated with the research institute. Naturally, a subset of objects of a given type, or a subtype, can be viewed as a result of a query on the set of all objects, which on very basic level illustrates how learning data mappings can be reduced to learning queries.

While basic mappings link objects of the same type, more general mappings define how the same type of information is represented in two different databases. For instance, the email address and the postal address of an individual may be represented in one way in the address book and in another way in the bibliographic databases, and naturally, the query asking for the email address and the postal address of a person identified by a given name will differ from one database to the other. While queries used in the context of linking objects of compatible types are essentially unary, queries used in the context of linking information are n-ary and we plan to approach inference of general database mappings by investigating and employing algorithms for inference of n-ary queries.

An important goal in this research is elaborating a formal definition of *learnability* (feasibility of inference) of a given class of concepts (schemas of queries). We plan to following the example of Gold (1967), which requires not only the existence of an efficient algorithm that infers concepts consistent with the given input but the ability to infer every concept from the given class with a sufficiently informative input. Naturally, learnability depends on two parameters. The first parameter is the class of concepts i.e., a class of schema and

a class of queries, from which the goal concept is to be inferred. The second parameter is the type of input that an inference algorithm is given. This can be a set of examples of a concept e.g., instances of RDF databases for which we wish to construct a schema or a selection of nodes that a goal query is to select. Alternatively, a more general interactive scenario can be used where the learning algorithm inquires the user about the goal concept e.g., by asking to indicate whether a given node is to be selected or not (as membership queries of Angluin (1987)). In general, the richer the input is, the richer class of concepts can be handled, however, the richer class of queries is to be handled, the higher computational cost is to be expected. The primary task is to find a good compromise and identify classes of concepts that are of high practical value, allow efficient inference with possibly simple type of input.

The main open problem for graph-shaped data studied by Links are how to infer queries, schemas, and schema-mappings for graph-structured data.

LOKI Project-Team

3. Research Program

3.1. Introduction

Interaction is by nature a dynamic phenomenon that takes place between interactive systems and their users. Redesigning interactive systems to better account for interaction requires fine understanding of these dynamics from the user side so as to better handle them from the system side. In fact, layers of actual interactive systems abstract hardware and system resources from a system and programing perspective. Following our Interaction Machine concept, we are reconsidering these architectures from the user's perspective, through different *levels of dynamics of interaction* (see Figure 1).

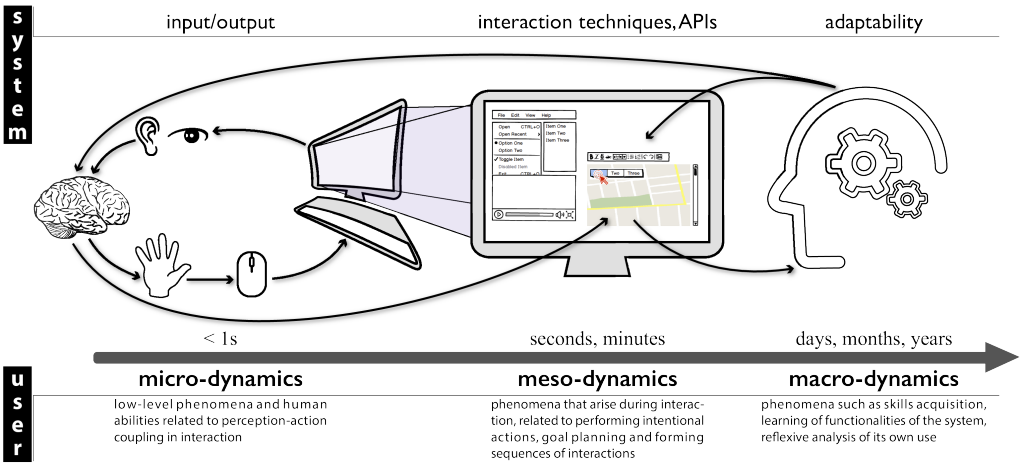

Figure 1. Levels of dynamics of interaction.

Considering phenomena that occur at each of these levels as well as their relationships will help us to acquire the necessary knowledge (Empowering Tools) and technological bricks (Interaction Machine) to reconcile the way interactive systems are designed and engineered with human abilities. Although our strategy is to investigate issues and address challenges for all of the three levels, our immediate priority is to focus on micro-dynamics since it concerns very fundamental knowledge about interaction and relates to very low-level parts of interactive systems, which is likely to influence our future research and developments at the other levels.

3.2. Micro-Dynamics

Micro-dynamics involve low-level phenomena and human abilities which are related to short time/instantness and to perception-action coupling in interaction, when the user has almost no control or consciousness of the action once it has been started. From a system perspective, it has implications mostly on input and output (I/O) management.

3.2.1. Transfer functions design and latency management

We have developed a recognized expertise in the characterization and the design of *transfer functions* [34], [45], i. e., the algorithmic transformations of raw user input for system use. Ideally, transfer functions should match the interaction context. Yet the question of how to maximize one or more criteria in a given context remains an open one, and on-demand adaptation is difficult because transfer functions are usually implemented at the lowest possible level to avoid latency. Latency has indeed long been known as a determinant of human performance in interactive systems [41] and recently regained attention with touch interactions [40]. These two problems require cross examination to improve performance with interactive systems: Latency can be a confounding factor when evaluating the effectiveness of transfer functions, and transfer functions can also include algorithms to compensate for latency.

We have recently proposed new cheap but robust methods for the measurement of end-to-end latency [2] and are currently working on compensation methods and the evaluation of their perceived side effects. Our goal is then to automatically adapt the transfer function to individual users and contexts of use while reducing latency in order to support stable and appropriate control. To achieve this, we will investigate combinations of low-level (embedded) and high-level (application) ways to take user capabilities and task characteristics into account and reduce or compensate for latency in different contexts, e. g., using a mouse or a touchpad, a touch-screen, an optical finger navigation device or a brain-computer interface. From an engineering perspective, this knowledge on low-level human factors will help us to rethink and redesign the I/O loop of interactive systems in order to better account for them and achieve more adapted and adaptable perception-action coupling.

3.2.2. Tactile feedback & haptic perception

We are also concerned with the physicality of human-computer interaction, with a focus on haptic perception and related technologies. For instance, when interacting with virtual objects such as software buttons on a touch surface, the user cannot feel the click sensation as with physical buttons. The tight coupling between how we perceive and how we manipulate objects is then essentially broken although this is instrumental for efficient direct manipulation. We have addressed this issue in multiple contexts by designing, implementing and evaluating novel applications of tactile feedback [5].

In comparison with many other modalities, one difficulty with tactile feedback is its diversity. It groups sensations of forces, vibrations, friction, or deformation. Although this is a richness, it also raises usability and technological challenges since each kind of haptic stimulation requires different kinds of actuators with their own parameters and thresholds. And results from one are hardly applicable to others. On a “knowledge” point of view, we want to better understand and empirically classify haptic variables and the kind of information they can represent (continuous, ordinal, nominal), their resolution, and their applicability to various contexts. From the “technology” perspective, we want to develop tools to inform and ease the design of haptic interactions taking best advantage of the different technologies in a consistent and transparent way.

3.3. Meso-Dynamics

Meso-dynamics relate to phenomena that arise during interaction, on a longer but still short time-scale. For users, it is related to performing intentional actions, to goal planning and tools selection, and to forming sequences of interactions based on a known set of rules or instructions. From the system perspective, it relates to how possible actions are exposed to the user and how they have to be executed (i. e., interaction techniques). It also has implication on the tools for designing and implementing those techniques (programming languages and APIs).

3.3.1. Interaction bandwidth and vocabulary

Interactive systems and their applications have an always-increasing number of available features and commands due to, e. g., the large amount of data to manipulate, increasing power and number of functionalities, or multiple contexts of use.

On the input side, we want to augment the *interaction bandwidth* between the user and the system in order to cope with this increasing complexity. In fact, most input devices capture only a few of the movements and actions the human body is capable of. Our arms and hands for instance have many degrees of freedom that are not fully exploited in common interfaces. We have recently designed new technologies to improve expressibility such as a bendable digitizer pen [36], or reliable technology for studying the benefits of finger identification on multi-touch interfaces [4].

On the output side, we want to expand users' *interaction vocabulary*. All of the features and commands of a system can not be displayed on screen at the same time and lots of *advanced* features are by default hidden to the users (e. g., hotkeys) or buried in deep hierarchies of command-triggering systems (e. g., menus). As a result, users tend to use only a subset of all the tools the system actually offers [44]. We will study how to help them to broaden their knowledge of available functions.

Through this "opportunistic" exploration of alternative and more expressive input methods and interaction techniques, we will particularly focus on the necessary technological requirements to integrate them into interactive systems, in relation with our redesign of the I/O stack at the micro-dynamics level.

3.3.2. Spatial and temporal continuity in interaction

At a higher-level, we will investigate how more expressive interaction techniques affect users' strategies when performing sequences of elementary actions and tasks. More generally, we will explore the "*continuity*" in interaction. Interactive systems have moved from one computer to multiple connected interactive devices (computer, tablets, phones, watches, etc.) that could also be augmented through a Mixed-Reality paradigm. This distribution of interaction raises new challenges from both usability and engineering perspectives that we clearly have to consider in our main objective of revisiting interactive systems [43]. It involves the simultaneous use of multiple devices and also the changes in the role of devices according to the location, the time, the task, and contexts of use: a tablet device can be used as the main device while traveling, and it becomes an input device or a secondary monitor for continuing the same task once in the office; a smart-watch can be used as a standalone device to send messages, but also as a remote controller for a wall-sized display. One challenge is then to design interaction techniques that support seamless and smooth transitions during these spatial and temporal changes of the system in order to maintain the continuity of uses and tasks, and how to integrate these principles in future interactive systems.

3.3.3. Expressive tools for prototyping, studying, and programming interaction

Current systems suffer from engineering issues that keep constraining and influencing how interaction is thought, designed, and implemented. Addressing the challenges we presented in this section and making the solutions possible require extended expressiveness, and researchers and designers must either wait for the proper toolkits to appear, or "hack" existing interaction frameworks, often bypassing existing mechanisms. For instance, numerous usability problems in existing interfaces stem from a common cause: the lack, or untimely discarding, of relevant information about how events are propagated and how changes come to occur in interactive environments. On top of our redesign of the I/O loop of interactive systems, we will investigate how to facilitate access to that information and also promote a more grounded and expressive way to describe and exploit input-to-output chains of events at every system level. We want to provide finer granularity and better-described connections between the *causes* of changes (e.g. input events and system triggers), their *context* (e.g. system and application states), their *consequences* (e.g. interface and data updates), and their *timing* [8]. More generally, a central theme of our Interaction Machine vision is to promote interaction as a first-class object of the system [33], and we will study alternative and better-adapted technologies for designing and programming interaction, such as we did recently to ease the prototyping of Digital Musical Instruments [1] or the programming of animations in graphical interfaces [10]. Ultimately, we want to propose a unified model of hardware and software scaffolding for interaction that will contribute to the design of our Interaction Machine.

3.4. Macro-Dynamics

Macro-dynamics involve longer-term phenomena such as skills acquisition, learning of functionalities of the system, reflexive analysis of its own use (e. g., when the user has to face novel or unexpected situations which require high-level of knowledge of the system and its functioning). From the system perspective, it implies to better support cross-application and cross-platform mechanisms so as to favor skill transfer. It also requires to improve the instrumentation and high-level logging capabilities to favor reflexive use, as well as flexibility and adaptability for users to be able to finely tune and shape their tools.

We want to move away from the usual binary distinction between "novices" and "experts" [3] and explore means to promote and assist digital skill acquisition in a more progressive fashion. Indeed, users have a permanent need to adapt their skills to the constant and rapid evolution of the tasks and activities they carry on a computer system, but also the changes in the software tools they use [47]. Software strikingly lacks powerful means of acquiring and developing these skills [3], forcing users to mostly rely on outside support (e. g., being guided by a knowledgeable person, following online tutorials of varying quality). As a result, users tend to rely on a surprisingly limited interaction vocabulary, or *make-do* with sub-optimal routines and tools [48]. Ultimately, the user should be able to master the interactive system to form durable and stabilized practices that would eventually become *automatic* and reduce the mental and physical efforts, making their interaction *transparent*.

In our previous work, we identified the fundamental factors influencing expertise development in graphical user interfaces, and created a conceptual framework that characterizes users' performance improvement with UIs [7], [3]. We designed and evaluated new command selection and learning methods to leverage user's digital skill development with user interfaces, on both desktop [6] and touch-based computers.

We are now interested in broader means to support the analytic use of computing tools:

- *to foster understanding of interactive systems.* As the digital world makes the shift to more and more complex systems driven by machine learning algorithms, we increasingly lose our comprehension of which process caused the system to respond in one way rather than another. We will study how novel interactive visualizations can help reveal and expose the "intelligence" behind, in ways that people better master their complexity.
- *to foster reflexion on interaction.* We will study how we can foster users' reflexion on their own interaction in order to encourage them to acquire novel digital skills. We will build real-time and off-line software for monitoring how user's ongoing activity is conducted at an application and system level. We will develop augmented feedbacks and interactive history visualization tools that will offer contextual visualizations to help users to better understand and share their activity, compare their actions to that of others, and discover possible improvement.
- *to optimize skill-transfer and tool re-appropriation.* The rapid evolution of new technologies has drastically increased the frequency at which systems are updated, often requiring to relearn everything from scratch. We will explore how we can minimize the cost of having to appropriate an interactive tool by helping users to capitalize on their existing skills.

We plan to explore these questions as well as the use of such aids in several contexts like web-based, mobile, or BCI-based applications. Although, a core aspect of this work will be to design systems and interaction techniques that will be as little platform-specific as possible, in order to better support skill transfer. Following our Interaction Machine vision, this will lead us to rethink how interactive systems have to be engineered so that they can offer better instrumentation, higher adaptability, and fewer separation between applications and tasks in order to support reuse and skill transfer.

MAGNET Project-Team

3. Research Program

3.1. Introduction

The main objective of MAGNET is to develop original machine learning methods for networked data in order to build applications like browsing, monitoring and recommender systems, and more broadly information extraction in information networks. We consider information networks in which the data consist of both feature vectors and texts. We model such networks as (multiple) (hyper)graphs wherein nodes correspond to entities (documents, spans of text, users, ...) and edges correspond to relations between entities (similarity, answer, co-authoring, friendship, ...). Our main research goal is to propose new online and batch learning algorithms for various problems (node classification / clustering, link classification / prediction) which exploit the relationships between data entities and, overall, the graph topology. We are also interested in searching for the best hidden graph structure to be generated for solving a given learning task. Our research will be based on generative models for graphs, on machine learning for graphs and on machine learning for texts. The challenges are the dimensionality of the input space, possibly the dimensionality of the output space, the high level of dependencies between the data, the inherent ambiguity of textual data and the limited amount of human labeling. An additional challenge will be to design scalable methods for large information networks. Hence, we will explore how sampling, randomization and active learning can be leveraged to improve the scalability of the proposed algorithms.

Our research program is organized according to the following questions:

1. How to go beyond vectorial classification models in Natural Language Processing (NLP) tasks?
2. How to adaptively build graphs with respect to the given tasks? How to create networks from observations of information diffusion processes?
3. How to design methods able to achieve a good trade-off between predictive accuracy and computational complexity?
4. How to go beyond strict node homophilic/similarity assumptions in graph-based learning methods?

3.2. Beyond Vectorial Models for NLP

One of our overall research objectives is to derive graph-based machine learning algorithms for natural language and text information extraction tasks. This section discusses the motivations behind the use of graph-based ML approaches for these tasks, the main challenges associated with it, as well as some concrete projects. Some of the challenges go beyond NLP problems and will be further developed in the next sections. An interesting aspect of the project is that we anticipate some important cross-fertilizations between NLP and ML graph-based techniques, with NLP not only benefiting from but also pushing ML graph-based approaches into new directions.

Motivations for resorting to graph-based algorithms for texts are at least threefold. First, online texts are organized in networks. With the advent of the web, and the development of forums, blogs, and micro-blogging, and other forms of social media, text productions have become strongly connected. Interestingly, NLP research has been rather slow in coming to terms with this situation, and most of the literature still focus on document-based or sentence-based predictions (wherein inter-document or inter-sentence structure is not exploited). Furthermore, several multi-document tasks exist in NLP (such as multi-document summarization and cross-document coreference resolution), but most existing work typically ignore document boundaries and simply apply a document-based approach, therefore failing to take advantage of the multi-document dimension [38], [41].

A second motivation comes from the fact that most (if not all) NLP problems can be naturally conceived as graph problems. Thus, NLP tasks often involve discovering a relational structure over a set of text spans (words, phrases, clauses, sentences, etc.). Furthermore, the *input* of numerous NLP tasks is also a graph; indeed, most end-to-end NLP systems are conceived as pipelines wherein the output of one processor is in the input of the next. For instance, several tasks take POS tagged sequences or dependency trees as input. But this structured input is often converted to a vectorial form, which inevitably involves a loss of information.

Finally, graph-based representations and learning methods appear to address some core problems faced by NLP, such as the fact that textual data are typically not independent and identically distributed, they often live on a manifold, they involve very high dimensionality, and their annotations is costly and scarce. As such, graph-based methods represent an interesting alternative to, or at least complement, structured prediction methods (such as CRFs or structured SVMs) commonly used within NLP. Graph-based methods, like label propagation, have also been shown to be very effective in semi-supervised settings, and have already given some positive results on a few NLP tasks [21], [43].

Given the above motivations, our first line of research will be to investigate how one can leverage an underlying network structure (e.g., hyperlinks, user links) between documents, or text spans in general, to enhance prediction performance for several NLP tasks. We think that a "network effect", similar to the one that took place in Information Retrieval (with the PageRank algorithm), could also positively impact NLP research. A few recent papers have already opened the way, for instance in attempting to exploit Twitter follower graph to improve sentiment classification [42].

Part of the challenge here will be to investigate how adequately and efficiently one can model these problems as instances of more general graph-based problems, such as node clustering/classification or link prediction discussed in the next sections. In a few cases, like text classification or sentiment analysis, graph modeling appears to be straightforward: nodes correspond to texts (and potentially users), and edges are given by relationships like hyperlinks, co-authorship, friendship, or thread membership. Unfortunately, modeling NLP problems as networks is not always that obvious. From the one hand, the right level of representation will probably vary depending on the task at hand: the nodes will be sentences, phrases, words, etc. From the other hand, the underlying graph will typically not be given a priori, which in turn raises the question of how we construct it. A preliminary discussion of the issue of optimal graph construction for semi-supervised learning in NLP is given in [21], [46]. We identify the issue of adaptive graph construction as an important scientific challenge for machine learning on graphs in general, and we will discuss it further in Section 3.3 .

As noted above, many NLP tasks have been recast as structured prediction problems, allowing to capture (some of the) output dependencies. How to best combine structured output and graph-based ML approaches is another challenge that we intend to address. We will initially investigate this question within a semi-supervised context, concentrating on graph regularization and graph propagation methods. Within such approaches, labels are typically binary or in a small finite set. Our objective is to explore how one propagates an exponential number of *structured labels* (like a sequence of tags or a dependency tree) through graphs. Recent attempts at blending structured output models with graph-based models are investigated in [43], [31]. Another related question that we will address in this context is how does one learn with *partial labels* (like partially specified tag sequence or tree) and use the graph structure to complete the output structure. This last question is very relevant to NLP problems where human annotations are costly; being able to learn from partial annotations could therefore allow for more targeted annotations and in turn reduced costs [33].

The NLP tasks we will mostly focus on are coreference resolution and entity linking, temporal structure prediction, and discourse parsing. These tasks will be envisioned in both document and cross-document settings, although we expect to exploit inter-document links either way. Choices for these particular tasks is guided by the fact that they are still open problems for the NLP community, they potentially have a high impact for industrial applications (like information retrieval, question answering, etc.), and we already have some expertise on these tasks in the team (see for instance [32], [28], [30]). As a midterm goal, we also plan to work on tasks more directly relating to micro-blogging, such as sentiment analysis and the automatic thread structuring of technical forums; the latter task is in fact an instance of rhetorical structure prediction [45].

We have already initiated some work on the coreference resolution with graph-based learning, by casting the problem as an instance of spectral clustering [30].

3.3. Adaptive Graph Construction

In most applications, edge weights are computed through a complex data modeling process and convey crucially important information for classifying nodes, making it possible to infer information related to each data sample even exploiting the graph topology solely. In fact, a widespread approach to several classification problems is to represent the data through an undirected weighted graph in which edge weights quantify the similarity between data points. This technique for coding input data has been applied to several domains, including classification of genomic data [40], face recognition [29], and text categorization [34].

In some cases, the full adjacency matrix is generated by employing suitable similarity functions chosen through a deep understanding of the problem structure. For example for the TF-IDF representation of documents, the affinity between pairs of samples is often estimated through the cosine measure or the χ^2 distance. After the generation of the full adjacency matrix, the second phase for obtaining the final graph consists in an edge sparsification/reweighting operation. Some of the edges of the clique obtained in the first step are pruned and the remaining ones can be reweighted to meet the specific requirements of the given classification problem. Constructing a graph with these methods obviously entails various kinds of loss of information. However, in problems like node classification, the use of graphs generated from several datasets can lead to an improvement in accuracy ([47], [22], [23]). Hence, the transformation of a dataset into a graph may, at least in some cases, partially remove various kinds of irregularities present in the original datasets, while keeping some of the most useful information for classifying the data samples. Moreover, it is often possible to accomplish classification tasks on the obtained graph using a running time remarkably lower than is needed by algorithms exploiting the initial datasets, and a suitable sparse graph representation can be seen as a compressed version of the original data. This holds even when input data are provided in a online/stream fashion, so that the resulting graph evolves over time.

In this project we will address the problem of adaptive graph construction towards several directions. The first one is about how to choose the best similarity measure given the objective learning task. This question is related to the question of metric and similarity learning ([24], [25]) which has not been considered in the context of graph-based learning. In the context of structured prediction, we will develop approaches where output structures are organized in graphs whose similarity is given by top-k outcomes of greedy algorithms.

A different way we envision adaptive graph construction is in the context of semi-supervised learning. Partial supervision can take various forms and an interesting and original setting is governed by two currently studied applications: detection of brain anomaly from connectome data and polls recommendation in marketing. Indeed, for these two applications, a partial knowledge of the information diffusion process can be observed while the network is unknown or only partially known. An objective is to construct (or complete) the network structure from some local diffusion information. The problem can be formalized as a graph construction problem from partially observed diffusion processes. It has been studied very recently in [36]. In our case, the originality comes either from the existence of different sources of observations or from the large impact of node contents in the network.

We will study how to combine graphs defined by networked data and graphs built from flat data to solve a given task. This is of major importance for information networks because, as said above, we will have to deal with multiple relations between entities (texts, spans of texts, ...) and also use textual data and vectorial data.

3.4. Prediction on Graphs and Scalability

As stated in the previous sections, graphs as complex objects provide a rich representation of data. Often enough the data is only partially available and the graph representation is very helpful in predicting the unobserved elements. We are interested in problems where the complete structure of the graph needs to be recovered and only a fraction of the links is observed. The link prediction problem falls into this category. We are also interested in the recommendation and link classification problems which can be seen as graphs

where the structure is complete but some labels on the links (weights or signs) are missing. Finally we are also interested in labeling the nodes of the graph, with class or cluster memberships or with a real value, provided that we have (some information about) the labels for some of the nodes.

The semi-supervised framework will be also considered. A midterm research plan is to study how graph regularization models help for structured prediction problems. This question will be studied in the context of NLP tasks, as noted in Section 3.2 , but we also plan to develop original machine learning algorithms that have a more general applicability. Inputs are networks whose nodes (texts) have to be labeled by structures. We assume that structures lie in some manifold and we want to study how labels can propagate in the network. One approach is to find a smooth labeling function corresponding to an harmonic function on both manifolds in input and output.

Scalability is one of the main issues in the design of new prediction algorithms working on networked data. It has gained more and more importance in recent years, because of the growing size of the most popular networked data that are now used by millions of people. In such contexts, learning algorithms whose computational complexity scales quadratically, or slower, in the number of considered data objects (usually nodes or edges, depending on the task) should be considered impractical.

These observations lead to the idea of using graph sparsification techniques in order to work on a part of the original network for getting results that can be easily extended and used for the whole original input. A sparsified version of the original graph can often be seen as a subset of the initial input, i.e. a suitably selected input subgraph which forms the training set (or, more in general, it is included in the training set). This holds even for the active setting. A simple example could be to find a spanning tree of the input graph, possibly using randomization techniques, with properties such that we are allowed to obtain interesting results for the initial graph dataset. We have started to explore this research direction for instance in [44].

At the level of mathematical foundations, the key issue to be addressed in the study of (large-scale) random networks also concerns the segmentation of network data into sets of independent and identically distributed observations. If we identify the data sample with the whole network, as it has been done in previous approaches [35], we typically end up with a set of observations (such as nodes or edges) which are highly interdependent and hence overly violate the classic i.i.d. assumption. In this case, the data scale can be so large and the range of correlations can be so wide, that the cost of taking into account the whole data and their dependencies is typically prohibitive. On the contrary, if we focus instead on a set of subgraphs independently drawn from a (virtually infinite) target network, we come up with a set of independent and identically distributed observations—namely the subgraphs themselves, where subgraph sampling is the underlying ergodic process [26]. Such an approach is one principled direction for giving novel statistical foundations to random network modeling. At the same time, because one shifts the focus from the whole network to a set of subgraphs, complexity issues can be restricted to the number of subgraphs and their size. The latter quantities can be controlled much more easily than the overall network size and dependence relationships, thus allowing to tackle scalability challenges through a radically redesigned approach.

Another way to tackle scalability problems is to exploit the inherent decentralized nature of very large graphs. Indeed, in many situations very large graphs are the abstract view of the digital activities of a very large set of users equipped with their own device. Nowadays, smartphones, tablets and even sensors have storage and computation power and gather a lot of data that serve to analytics, prediction, suggestion and personalized recommendation. Gathering all user data in large data centers is costly because it requires oversized infrastructures with huge energy consumption and large bandwidth networks. Even though cloud architectures can optimize such infrastructures, data concentration is also prone to security leaks, lost of privacy and data governance for end users. The alternative we have started to develop in Magnet is to devise decentralized, private and personalized machine learning algorithms so that they can be deployed in the personal devices. The key challenges are therefore to learn in a collaborative way in a network of learners and to preserve privacy and control on personal data.

3.5. Beyond Homophilic Relationships

In many cases, algorithms for solving node classification problems are driven by the following assumption: linked entities tend to be assigned to the same class. This assumption, in the context of social networks, is known as homophily ([27], [37]) and involves ties of every type, including friendship, work, marriage, age, gender, and so on. In social networks, homophily naturally implies that a set of individuals can be parted into subpopulations that are more cohesive. In fact, the presence of homogeneous groups sharing common interests is a key reason for affinity among interconnected individuals, which suggests that, in spite of its simplicity, this principle turns out to be very powerful for node classification problems in general networks.

Recently, however, researchers have started to consider networked data where connections may also carry a negative meaning. For instance, disapproval or distrust in social networks, negative endorsements on the Web. Although the introduction of signs on graph edges appears like a small change from standard weighted graphs, the resulting mathematical model, called signed graphs, has an unexpectedly rich additional complexity. For example, their spectral properties, which essentially all sophisticated node classification algorithms rely on, are different and less known than those of graphs. Signed graphs naturally lead to a specific inference problem that we have discussed in previous sections: link classification. This is the problem of predicting signs of links in a given graph. In online social networks, this may be viewed as a form of sentiment analysis, since we would like to semantically categorize the relationships between individuals.

Another way to go beyond homophily between entities will be studied using our recent model of hypergraphs with bipartite hyperedges [39]. A bipartite hyperedge connects two ends which are disjoint subsets of nodes. Bipartite hyperedges is a way to relate two collections of (possibly heterogeneous) entities represented by nodes. In the NLP setting, while hyperedges can be used to model bags of words, bipartite hyperedges are associated with relationships between bags of words. But each end of bipartite hyperedges is also a way to represent complex entities, gathering several attribute values (nodes) into hyperedges viewed as records. Our hypergraph notion naturally extends directed and undirected weighted graph. We have defined a spectral theory for this new class of hypergraphs and opened a way to smooth labeling on sets of nodes. The weighting scheme allows to weigh the participation of each node to the relationship modeled by bipartite hyperedges accordingly to an equilibrium condition. This condition provides a competition between nodes in hyperedges and allows interesting modeling properties that go beyond homophily and similarity over nodes (the theoretical analysis of our hypergraphs exhibits tight relationships with signed graphs). Following this competition idea, bipartite hyperedges are like matches between two teams and examples of applications are team creation. The basic tasks we are interested in are hyperedge classification, hyperedge prediction, node weight prediction. Finally, hypergraphs also represent a way to summarize or compress large graphs in which there exists highly connected couples of (large) subsets of nodes.