



IN PARTNERSHIP WITH:  
**Ecole des Mines de Nantes**  
**Université Nantes**

Activity Report 2016

## **Project-Team TASC**

Theory, Algorithms and Systems for  
Constraints

IN COLLABORATION WITH: Laboratoire d'Informatique de Nantes Atlantique (LINA)

RESEARCH CENTER  
**Rennes - Bretagne-Atlantique**

THEME  
**Architecture, Languages and Compila-  
tion**



## Table of contents

<b>1. Members</b>	<b>1</b>
<b>2. Overall Objectives</b>	<b>2</b>
2.1.1. Origin and Current Situation	2
2.1.2. General Challenges	2
<b>3. Research Program</b>	<b>3</b>
3.1. Overview	3
3.2. Fundamental Research Topics	3
3.2.1. Constraints Classification, Reformulation and Filtering	3
3.2.2. Convergence between Discrete and Continuous	4
3.2.3. Dynamic, Interactive and over Constrained Problems	5
3.2.4. Solvers	5
<b>4. Application Domains</b>	<b>5</b>
4.1. Introduction	5
4.2. Panorama	6
<b>5. Highlights of the Year</b>	<b>6</b>
<b>6. New Software and Platforms</b>	<b>6</b>
6.1. AIUR	6
6.2. CHOCO	7
6.3. GCCat	7
6.4. GCCat on time series	7
6.5. GHOST	9
6.6. TorchCraft	9
<b>7. New Results</b>	<b>10</b>
7.1. Discrete Convexity	10
7.2. Transducers	10
7.3. Compositional Glue Matrix and Bound for Time-Series Constraints	10
7.4. Reformulation of time-series constraint in MIP	11
7.5. Scheduling Constraint for Video Summarisation	13
7.6. A Model Seeker for Learning Constraints Models from Positive Samples	13
7.7. Global Constraint Catalog Volume II: Time-Series Constraints	13
7.8. Probabilistic Model for Binary CSP	15
7.9. Estimating parallel runtimes for randomized algorithms in constraint solving	15
7.10. ghost: A Combinatorial Optimization Framework for Real-Time Problems	16
7.11. TorchCraft: a Library for Machine Learning Research on Real-Time Strategy Games	16
7.12. POSL: A Parallel-Oriented metaheuristic-based Solver Language	16
7.13. Towards Automated Strategies in Satisfiability Modulo Theory	16
7.14. Using CP for the Urban Transit Crew Rescheduling Problem	17
7.15. Traveling salesman and the tree: the importance of search in CP	17
7.16. Event Selection Rules to Compute Explanations	17
7.17. Towards energy-proportional Clouds partially powered by renewable energy	18
<b>8. Bilateral Contracts and Grants with Industry</b>	<b>18</b>
8.1. Bilateral Contracts with Industry	18
8.2. Bilateral Grants with Industry	18
<b>9. Partnerships and Cooperations</b>	<b>18</b>
9.1. Regional Initiatives	18
9.1.1. EPOC	18
9.1.2. SmartCat	19
9.1.3. Atlanstic 1	19
9.1.4. Atlanstic 2	19

9.2. National Initiatives	19
9.3. European Initiatives	20
9.4. International Initiatives	20
9.5. International Research Visitors	21
9.5.1. Visits of International Scientists	21
9.5.2. Visits to International Teams	21
<b>10. Dissemination</b> .....	<b>21</b>
10.1. Promoting Scientific Activities	21
10.1.1. Leadership within the Scientific Community	21
10.1.2. Research Administration	21
10.2. Teaching - Supervision - Juries	21
10.3. Popularization	22
<b>11. Bibliography</b> .....	<b>22</b>

# Project-Team TASC

*Creation of the Project-Team: 2011 January 01, end of the Project-Team: 2016 December 31*

## Keywords:

### Computer Science and Digital Science:

- 2.1.5. - Constraint programming
- 3.2.1. - Knowledge bases
- 3.2.3. - Inference
- 6.1. - Mathematical Modeling
- 7.2. - Discrete mathematics, combinatorics
- 8.2. - Machine learning
- 8.6. - Decision support

### Other Research Topics and Application Domains:

- 4.1. - Fossile energy production (oil, gas)
- 4.3. - Renewable energy production
- 8.3. - Urbanism and urban planning

## 1. Members

### Faculty Members

Nicolas Beldiceanu [Team leader, EMN, Professor, HDR]  
Mohamed Amine Balafrej [EMN, Associate Professor, until Sep 2016]  
Frédéric Benhamou [Univ. Nantes, Professor]  
Xavier Lorca [EMN, Associate Professor, HDR]  
Eric Monfroy [Univ. Nantes, Professor, HDR]  
Thierry Petit [EMN, Associate Professor, HDR]  
Florian Richoux [Univ. Nantes, Associate Professor]  
Gilles Simonin [EMN, Associate Professor, from Sep 2016]  
Charlotte Truchet [Univ. Nantes, Associate Professor]

### Engineer

Charles Prud'Homme [ARMINES]

### PhD Students

Ekaterina Arafailova [ARMINES]  
Anicet Bart [Univ. Nantes]  
Gilles Madi Wamba [EMN]  
Alejandro Reyes Amaro [Univ. Nantes, until Sep 2016]  
Ignacio Salas Donoso [EMN, until Apr 2016]

### Post-Doctoral Fellow

Abbas El Dor [ARMINES]

### Administrative Assistant

Anne Claire Binetruy [Inria]

## 2. Overall Objectives

### 2.1. Objectives of the team

#### 2.1.1. Origin and Current Situation

Constraint programming emerges in the eighties and develops at **the intersection of Artificial Intelligence and Operations Research**, of Computer Science and Mathematics. Multidisciplinary by nature it keeps on using knowledge from various topics such as discrete mathematics, theoretical computer science (graph theory, combinatorics, algorithmic, complexity), functional analysis and optimization, IT and software engineering. Constraint programming was identified in 1996 by the ACM as a *strategic topic for Computer Science*. The turn of the century has seen the development of optimization technology in the industry (with notably Ilog, IBM, Dash and more recently Microsoft, Google and Dynadec) and the corresponding scientific field, at the border of Constraint Programming, Mathematical Programming, Local Search and Numerical Analysis. Optimisation technology is now assisting public sector, companies and people to some extent for making decisions that use resources better and match specific requirements in an increasingly complex world. Indeed, computer aided decision and optimization is becoming one of the cornerstones for providing assistance to all kinds of human activities.

Today, with the preeminence of optimization technology in most industrial sectors, we argue that quick and ad hoc solutions, often used today, cannot support the long-term development of optimization technology and its broad diffusion. We also argue that there should be a much more direct link between mathematical results and their systematic reuse in the main fields of optimization technology.

#### 2.1.2. General Challenges

In spite of its importance, computer aided decision and optimization suffers from a number of fundamental weaknesses that prevent from taking advantage of its full potential and hinder its progress and its capacity to deal with more and more complex situations. This can be mostly blamed on the diversity of actors, which are:

- Spread out in distinct scientific communities, each with its own focus:
  - On the one hand, computer science for providing languages, modelling tools and libraries. While focusing on providing flexible and powerful programming paradigm that can be easily deployed and maintained on modern architectures, it does not address the central question of how to come up in a systematic way with efficient methods for optimization and decision problems.
  - On the other hand, applied mathematics for the theory part. The focus is to come up with powerful abstractions that allow understanding the structure of a class of problems, independently of its practical and systematic uses in modern software components.
- Spread out in distinct technological communities, each independently pushing its own solving paradigm like constraint programming, linear and integer programming, continuous optimization, constraint-based local search (e.g., **COMET**). To some extent, most of these techniques exploit in different ways the same mathematical results, that are manually adapted to fit the main way to proceed of a given technology.

Thus, a first challenge encountered by constraint programming is the design of computer systems implementing **in a transparent way** effective solving techniques.

- Ideally, the user must be able to **describe his problem in a high level modelling language** without being concerned with the underlying solving mechanisms used. Such systems must also be independent both from any computer programming language and from any resolution engine.
- In order to assist user, systems must also offer **digital knowledge base in problem solving** that make available state of the art models and heuristics for large set of well identified problems.
- Lastly, the user must have the ability to interpret the returned solutions, in particular within the context of **over constrained problems where it is necessary to partly relax some constraints**, and that in the most realistic possible way.

A second challenge resides in the **speed of resolution especially in the context of large-scale data**. One has to adapt techniques such as generic consistency algorithms, graph algorithms, mathematical programming, meta-heuristics and to integrate them within the framework of constraint programming. This integration generates new questions such as the design of incremental algorithms, the automatic decomposition or the automatic reformulation of problems.

Finally a third challenge deals with the use of constraint programming in the context of **complex industrial problems**, especially when both discrete and continuous aspects are present. Complexity has multiple causes such as:

- the combination of temporal and spatial aspects, of continuous and discrete aspects,
- the dynamic character of some phenomena inducing a modification of the constraints and data during time,
- the difficulty of expressing some physical constraints, e.g. load balancing and temporal stability,
- the necessary decomposition of large problems inducing significant solution performance losses.

## 3. Research Program

### 3.1. Overview

Basic research is guided by the challenges raised before: to classify and enrich the models, to automate reformulation and resolution, to dissociate declarative and procedural knowledge, to come up with theories and tools that can handle problems involving both continuous and discrete variables, to develop modelling tools and to come up with solving tools that scale well. On the one hand, **classification aspects** of this research are integrated within a knowledge base about combinatorial problem solving: the global constraint catalog (see <http://sofdem.github.io/gccat/>). On the other hand, **solving aspects** are capitalized within the constraint solving system **CHOCO**. Lastly, within the framework of its activities of valorisation, teaching and of partnership research, the team uses constraint programming for solving various concrete problems. The challenge is, on one side to increase the visibility of the constraints in the others disciplines of computer science, and on the other side to contribute to a broader diffusion of the constraint programming in the industry.

### 3.2. Fundamental Research Topics

This part presents the research topics investigated by the project:

- Global Constraints Classification, Reformulation and Filtering,
- Convergence between Discrete and Continuous,
- Dynamic, Interactive and over Constrained Problems,
- Solvers.

These research topics are in fact not independent. The work of the team thus frequently relates transverse aspects such as explained global constraints, Benders decomposition and explanations, flexible and dynamic constraints, linear models and relaxations of constraints.

#### 3.2.1. Constraints Classification, Reformulation and Filtering

In this context our research is focused (a) first on identifying recurring combinatorial structures that can be used for modelling a large variety of optimization problems, and (b) exploit these combinatorial structures in order to come up with efficient algorithms in the different fields of optimization technology. The key idea for achieving point (b) is that many filtering algorithms both in the context of Constraint Programming, Mathematical Programming and Local Search can be interpreted as the maintenance of invariants on specific domains (e.g., graph, geometry). The systematic classification of **global constraints** and of their relaxation brings a synthetic view of the field. It establishes links between the properties of the concepts used to describe constraints and the properties of the constraints themselves. Together with **SICS**, the team develops and maintains

a catalog of global constraints, which describes the semantics of more than 431 constraints, and proposes a unified mathematical model for expressing them. This model is based on graphs, automata and logic formulae and allows to derive filtering methods and automatic reformulation for each constraint in a unified way (see <http://www.emn.fr/x-info/sdemasse/gccat/index.html>). We consider hybrid methods (i.e., methods that involve more than one optimization technology such as constraint programming, mathematical programming or local search), to draw benefit from the respective advantages of the combined approaches. More fundamentally, the study of hybrid methods makes it possible to compare and connect strategies of resolution specific to each approach for then conceiving new strategies. Beside the works on classical, complete resolution techniques, we also investigate local search techniques from a mathematical point of view. These partly random algorithms have been proven very efficient in practice, although we have little theoretical knowledge on their behaviour, which often makes them problem-specific. Our research in that area is focused on a probabilistic model of local search techniques, from which we want to derive quantified information on their behaviour, in order to use this information directly when designing the algorithms and exploit their performances better. We also consider algorithms that maintain local and global consistencies, for more specific models. Having in mind the trade off between genericity and effectiveness, the effort is put on the efficiency of the algorithms with guarantee on the produced levels of filtering. This effort results in adapting existing techniques of resolution such as graph algorithms. For this purpose we identify necessary conditions of feasibility that can be evaluated by efficient incremental algorithms. Genericity is not neglected in these approaches: on the one hand the constraints we focus on are applicable in many contexts (for example, graph partitioning constraints can be used both in logistics and in phylogeny); on the other hand, this work led to study the portability of such constraints and their independence with specific solvers. This research orientation gathers various work such as strong local consistencies, graph partitioning constraints, geometrical constraints, and optimization and soft constraints. Within the perspective to deal with complex industrial problems, we currently develop meta constraints (e.g. *geost*) handling all together the issues of large-scale problems, dynamic constraints, combination of spatial and temporal dimensions, expression of business rules described with first order logic.

### 3.2.2. Convergence between Discrete and Continuous

Many industrial problems mix continuous and discrete aspects that respectively correspond to physical (e.g., the position, the speed of an object) and logical (e.g., the identifier, the nature of an object) elements. Typical examples of problems are for instance:

- *Geometrical placement problems* where one has to place in space a set of objects subject to various geometrical constraints (i.e., non-overlapping, distance). In this context, even if the positions of the objects are continuous, the structure of optimal configurations has a discrete nature.
- *Trajectory and mission planning problems* where one has to plan and synchronize the moves of several teams in order to achieve some common goal (i.e., fire fighting, coordination of search in the context of rescue missions, surveillance missions of restricted or large areas).
- *Localization problems in mobile robotic* where a robot has to plan alone (only with its own sensors) its trajectory. This kind of problematic occurs in situations where the GPS cannot be used (e.g., under water or Mars exploration) or when it is not precise enough (e.g., indoor surveillance, observation of contaminated sites).

Beside numerical constraints that mix continuous and integer variables we also have global constraints that involve both type of variables. They typically correspond to graph problems (i.e., graph colouring, domination in a graph) where a graph is dynamically constructed with respect to geometrical and-or temporal constraints. In this context, the key challenge is avoiding decomposing the problem in a discrete and continuous parts as it is traditionally the case. As an illustrative example consider *the wireless network deployment problem*. On the one hand, the continuous part consists of finding out where to place a set of antenna subject to various geometrical constraints. On the other hand, by building an interference graph from the positions of the antenna, the discrete part consists of allocating frequencies to antenna in order to avoid interference. In the context of convergence between discrete and continuous variables, our goals are:

- First to identify and compare typical class of techniques that are used in the context of continuous and discrete solvers.



- To see how one can unify and/or generalize these techniques in order to handle in an integrated way continuous and discrete constraints within the same framework.

### 3.2.3. Dynamic, Interactive and over Constrained Problems

Some industrial applications are defined by a set of constraints which may change over time, for instance due to an interaction with the user. Many other industrial applications are over-constrained, that is, they are defined by set of constraints which are more or less important and cannot be all satisfied at the same time. Generic, dedicated and explanation-based techniques can be used to deal efficiently with such applications. Especially, these applications rely on the notion of *soft constraints* that are allowed to be (partially) violated. The generic concept that captures a wide variety of soft constraints is the violation measure, which is coupled with specific resolution techniques. Lastly, soft constraints allow to combine the expressive power of global constraints with local search frameworks.

### 3.2.4. Solvers

- *Discrete solver* Our theoretical work is systematically validated by concrete experimentations. We have in particular for that purpose the **CHOCO** constraint platform. The team develops and maintains **CHOCO** initially with the assistance of the laboratory e-lab of Bouygues (G. Rochart), the company Amadeus (F. Laburthe), and others researchers such as **N. Jussien** and **H. Cambazard** (4C, INP Grenoble). Since 2008 the main developments are done by **Charles Prud'homme** and **Xavier Lorca**. The functionalities of **CHOCO** are gradually extended with the outcomes of our works: design of constraints, analysis and visualization of explanations, etc. The open source **CHOCO** library is downloaded on average 450 times each month since 2006. **CHOCO** is developed in line with the research direction of the team, in an open-minded scientific spirit. Contrarily to other solvers where the efficiency often relies on problem-specific algorithms, **CHOCO** aims at providing the users both with reusable techniques (based on an up-to-date implementation of the **global constraint catalogue**) and with a variety of tools to ease the use of these techniques (clear separation between model and resolution, event-based solver, management of the over-constrained problems, explanations, etc.).
- *Discrete continuous* We use discrete convexity to describe filtering for families of constraints: We introduce a propagator for pairs of Sum constraints, where the expressions in the sums respect a form of convexity. This propagator is parametric and can be instantiated for various concrete pairs, including Deviation, Spread, and the conjunction of  $\text{Linear} \leq$  and Among. We show that despite its generality, our propagator is competitive in theory and practice with state-of-the-art propagators.
- *Constraint programming and verification* Constraint Programming has already had several applications to verification problems. It also has many common ideas with Abstract Interpretation, a theory of approximation of the semantics of programs. In both cases, we are interested in a particular set (solutions in CP, program traces in semantics), which is hard or impossible to compute, and this set is replaced by an over-approximation (consistent domains / abstract domains). Previous works (internship of Julie Laniau, PhD of **Marie Pelleau**, collaboration with the Abstract Interpretation team at the ENS and **Antoine Miné** in particular) have exhibited some of these links, and identified some situations where the two fields, Abstract Interpretation and Constraint Programming, can complement each other. It is the case in real-time stream processing languages, where Abstract Interpretation techniques may not be precise enough when analyzing loops. With the PhD of **Anicet Bart**, we are currently working on using CP techniques to find loop invariants for the **Faust language**, a functional sound processing language.

This work around the design and the development of solvers thus forms the fourth direction of basic research of the project.

## 4. Application Domains

### 4.1. Introduction

Constraint programming deals with the resolution of decision problems by means of rational, logical and computational techniques. Above all, constraint programming is founded on a clear distinction between, on the one hand the description of the constraints intervening in a problem, and on the other hand the techniques used for the resolution. The ability of constraint programming to handle in a flexible way heterogeneous constraints has raised the commercial interest for this paradigm in the early eighties. Among his fields of predilection, one finds traditional applications such as computer aided decision-making, scheduling, planning, placement, logistics or finance, as well as applications such as electronic circuits design (simulation, checking and test), DNA sequencing and phylogeny in biology, configuration of manufacturing products or web sites, formal verification of code.

## 4.2. Panorama

In 2015 the **TASC** team was involved in the following application domains:

- *Replanning* in industrial timetabling problems in a Labcom project with **Eurodécision** (see Figure 9).
- *Planning and replanning* in Data Centres taking into account energy consumption in the EPOC (Energy Proportional and Opportunistic Computing system) project.
- *Packing complex shapes* in the context of a warehouse (NetWMS2 project).
- Building decision support system for *resilient city development planning wrt climat change* (**GRACeFUL** project).
- *Optimizing electricity production* in the Gaspard Monge call program for Optimisation and Operation Research in the context of electricity production. In 2015 we were focussing on the systematic reformulation of time-series constraints for MIP solvers. This was done in order to integrate time-series constraints in existing integer linear programming models for electricity production.

## 5. Highlights of the Year

### 5.1. Highlights of the Year

#### 5.1.1. Awards

Award at the **MiniZinc Challenge 2016 solver competition** in the Fixed category (Bronze). The aim of the challenge is to start to compare various constraint solving technology on the same problems sets. The focus is on finite domain propagation solvers. An auxiliary aim is to build up a library of interesting problem models, which can be used to compare solvers and solving technologies.

## 6. New Software and Platforms

### 6.1. AIUR

(Artificial Intelligence Using Randomness)  
FUNCTIONAL DESCRIPTION

The main idea is to be unpredictable by making some stochastic choices. The AI starts a game with a "mood" randomly picked up among 5 moods, dictating some behaviors (aggressive, fast expand, macro-game, ...). In addition, some other choices (productions, timing attacks, early aggressions, ...) are also taken under random conditions.

Learning is an essential part of AIUR . For this, it uses persistent I/O files system to record which moods are efficient against a given opponent, in order to modify the probability distribution for the mood selection. The current system allows both on-line and off-line learning.

- Contact: Florian Richoux
- URL: <https://github.com/AIUR-group/AIUR>

## 6.2. CHOCO

**KEYWORDS:** Constraint Programming - Scheduling - Optimisation - Operational research - Financial analysis - Planning

**SCIENTIFIC DESCRIPTION**

or second consecutive year, CHOCO has participated at the MiniZinc Challenge , an annual competition of constraint programming solvers. In concurrency with 16 other solvers, CHOCO has won three bronze medals in three out of four categories (Free search, Parallel search and Open class). Five versions have been released all year long, the last one (v3.3.0, Dec. 17th) has the particularity to be promoted on Maven Central Repository. The major modifications were related to a simplification of the API but also improvement of the overall solver.

Within the context of the PhD thesis of Charles Prud'homme, a domain specific language that allows prototyping propagation engines was integrated within CHOCO, A paper appears at Constraints.

Within the context of the PhD thesis of Charles Prud'homme, a generic strategy based on explanations for large neighborhood search was designed and integrated within CHOCO. A corresponding paper appears at Constraints.

Within the context of the PhD thesis of Jean-Guillaume Fages, a documented package for graph variables was designed and integrated within CHOCO .

**FUNCTIONAL DESCRIPTION**

CHOCO is a Java discrete constraints library for describing hard combinatorial problems in the form of Constraint Satisfaction Problems and solving them with Constraint Programming techniques. Choco can be used to solve a broad range of real combinatorial problems. It is easy to use and offers excellent performance. This technique enables non-specialists to tackle strategic or operational problems, for instance, problems related to planning, scheduling, logistics, financial analysis and bio-informatics.

- Participants: Charles Prud'homme, Nicolas Beldiceanu, Jean-Guillaume Fages, Xavier Lorca, Thierry Petit and Rémi Douence
- Partner: Ecole des Mines de Nantes
- Contact: Julien Prud'homme
- URL: <http://www.choco-solver.org/>

## 6.3. GCCat

Global Constraint Catalog

**KEYWORDS:** Constraint Programming - Graph - Global constraint

**FUNCTIONAL DESCRIPTION**

The global constraint catalog presents and classifies global constraints and describes different aspects with meta data.

- Participants: Nicolas Beldiceanu and Sophie Demassey
- Contact: Nicolas Beldiceanu
- URL: <http://sofdem.github.io/gccat/gccat/index.html>

## 6.4. GCCat on time series

Global Constraint Catalog, Volume II, time-series constraints

**KEYWORDS:** Constraint Programming - Sequence - Transducer - Global constraint

**FUNCTIONAL DESCRIPTION**

The second volume of the Global Constraint Catalogue is devoted to time-series constraints. Within the context of Constraint Programming, time-series constraints go back to the work of Goldin and Kanellakis. This volume contains 626 constraints, which are explicitly described in terms of automata with accumulators. Checkers and propagators for all these constraints were synthesised from 22 transducers.

As in the first volume, the global constraints described in this second volume are not only accessible to humans, who can read the catalogue when searching for some information. It is also available to machines, which can read and interpret it. This is why there also exists an electronic version of this catalogue where one can get, for all time-series constraints, a complete description in terms of meta-data used in the first volume. In fact, unlike the first volume, *all the meta-data* of the electronic version as well as *all text and figures* of this second volume were automatically generated. While this second volume is by no means supposed to contain all possible time-series constraints, it contributes in the context of time-series constraints to the *systematic reconstruction* of the Global Constraint Catalogue that we have previously advocated. This reconstruction is based on the following methodology:

- First reuse, adapt or come up with abstractions, which allow to concisely represent structures and properties of time series as abstract combinatorial objects. In our context these abstractions essentially correspond to:
  1. Transducers where letters of the output alphabet are interpreted as semantic letters indicating how to recognise pattern occurrences.
  2. Transducers glue matrices expressing the relationship between the prefix, the suffix and the full sequence passed to a transducer.
  3. Properties associated to regular expressions corresponding to fragments of the input language of our transducers.
- Second, create from these abstract combinatorial objects a data base of concrete combinatorial objects.
- Third, synthesise concrete code for various technologies, languages, tasks from this data base of concrete combinatorial objects. In this context, correctness and efficiency of the synthesised code are essentially side product of:
  - The correctness of the formulae of our data base which is itself based on the wellformedness of our abstractions.
  - The generality behind our abstract combinatorial objects.

The time-series catalogue is done in the following way:

- All time-series constraints are now defined in a *compositional way* from a few basic constituents, i.e., patterns, features, aggregators, and predicates, which completely define the meaning of a constraint, where patterns are defined using regular expressions.
- Constraint names are now constructed in a systematic way as the *concatenation* of pattern name, feature name, and aggregation or predicate name.
- Given a pattern  $p$ , checkers and constraints are now *systematically synthesised* from a transducer that, given an input sequence over the input alphabet  $\{<, =, >\}$ , compares two adjacent values of a time-series and determines an output sequence over a output semantic alphabet describing how to recognise the occurrences of  $p$ .
- For each time-series constraint associated with a pattern  $p$ , the generation of an automaton with accumulators is completely driven by the transducer associated with pattern  $p$  as well as by *decoration tables* describing for each semantic letter of the output alphabet of the transducers how to generate accumulator updates. Code optimisation is ensured by using decoration tables that depend on properties of the pattern, of the feature, and of the aggregator associated with the time-series constraint.
- Lower and upper bounds of characteristics of time-series that appear in the restriction slot of a time-series constraint are synthesised from a *few parameterised formulae* that only depend on a restricted set of characteristics of the regular expression associated with the pattern.
- Parametrised glue matrices are provided for each transducer that corresponds to reversible time-series constraints. A concrete glue matrix is given for each reversible time-series constraint.

- Linear invariants are systematically obtained by applying the Farkas Lemma to the automata with accumulators that were synthesised. They consist of *linear constraints typically linking consecutive accumulator values*, e.g., see the legend of the second automaton of the constraints, which are generated even with non-linear accumulator updates. Missing linear invariants will be completed later on.
- Last but not least, time-series constraints were used for generating time-series verifying a conjunction of constraints both in the context of Constraint Programming and in the context of Linear Programming.
- In the context of sequential pattern mining, time-series constraint checkers can be used to identify and extract patterns from fixed sequences. While the time-series catalogue may need to be extended in order to capture more patterns, having a possibly large set of fixed time-series constraints is a natural safeguard to prevent overfitting when dealing with few sequences, at a price of not finding patterns that are not covered by the catalogue.
- Finally, both SICStus and MiniZinc code are synthesised. The later allows using time series constraints on many plate forms such as Choco, Gecode, ORtools, Cplex or Gurobi and is available Electronic Constraint Catalogue.
- Participants: Ekaterina Arafailova, Nicolas Beldiceanu, Rémi Douence, Mats Carlsson, Pierre Flener, Maria Andreina Francisco Rodriguez, Justin Pearson, Helmut Simonis
- Contact: Nicolas Beldiceanu
- URL: <https://arxiv.org/abs/1609.08925>

## 6.5. GHOST

General meta-Heuristic Optimization Solving Tool

FUNCTIONAL DESCRIPTION

GHOST is a template C++ library designed for StarCraft:BroodWartm. GHOST implements a meta-heuristic solver aiming to solve any kind of combinatorial and optimization RTS-related problems represented by a csp /cop. The solver handles dedicated geometric and assignment constraints in a way that is compatible with very strong real time requirements.

- Contact: Florian Richoux
- URL: <http://github.com/richoux/GHOST>

## 6.6. TorchCraft

Machine learning framework for games

FUNCTIONAL DESCRIPTION

TorchCraft is a library that enables deep learning research on Real-Time Strategy (RTS) games such as StarCraft: Brood War, by making it easier to control these games from a machine learning framework, here Torch. This white paper argues for using RTS games as a benchmark for AI research, and describes the design and components of TorchCraft.

- Participants: Gabriel Synnaeve, Nantas Nardelli, Alex Auvolat, Soumith Chintala, Timothée Lacroix, Zeming Lin, Florian Richoux, Nicolas Usunier
- Contact: Florian Richoux
- URL: <https://arxiv.org/abs/1611.00625>

## 7. New Results

### 7.1. Discrete Convexity

We introduce a propagator for pairs of Sum constraints, where the expressions in the sums respect a form of convexity. This propagator is parametric and can be instantiated for various concrete pairs, including Deviation, Spread, and the conjunction of Linear( $\leq$ ) and Among. We show that despite its generality, our propagator (see Figure 1) is competitive in theory and practice with state-of-the-art propagators. (see [AI journal paper](#)).

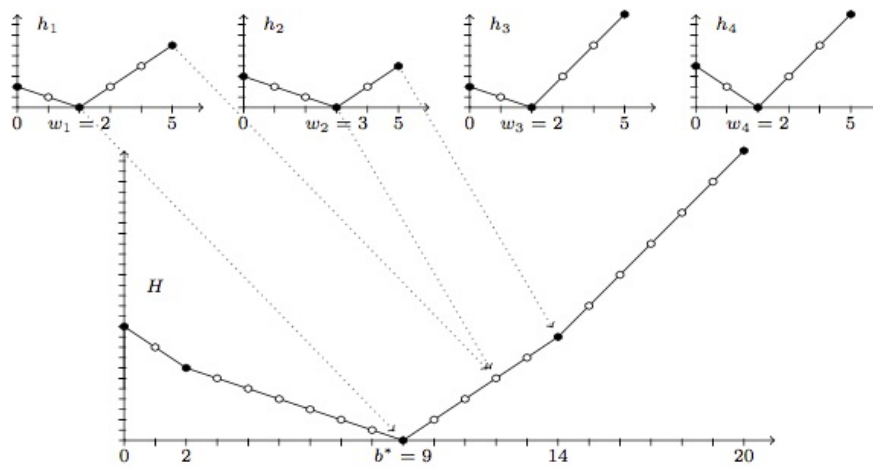


Figure 1. Illustration of the filtering wrt  $h$  function

### 7.2. Transducers

We describe a large family of constraints for structural time series by means of function composition. These constraints are on aggregations of features of patterns that occur in a time series, such as the number of its peaks, or the range of its steepest ascent. The patterns and features are usually linked to physical properties of the time series generator, which are important to capture in a constraint model of the system, i.e. a conjunction of constraints that produces similar time series. We formalise the patterns using finite transducers, whose output alphabet corresponds to semantic values that precisely describe the steps for identifying the occurrences of a pattern. Based on that description, we automatically synthesise automata with accumulators, as well as constraint checkers. The description scheme not only unifies the structure of the existing 30 time-series constraints in the Global Constraint Catalogue, but also leads to over 600 new constraints, with more than 100,000 lines of synthesised code. (see [Constraint journal paper](#))

### 7.3. Compositional Glue Matrix and Bound for Time-Series Constraints

Integer time series are often subject to constraints on the aggregation of the integer features of all occurrences of some pattern within the series. For example, the number of inflexions may be constrained, or the sum of the peak maxima, or the minimum of the peak widths. It is currently unknown how to maintain domain consistency efficiently on such constraints. We propose parametric ways of systematically deriving glue constraints (see Figures 3 and 4 for the parametric and concrete glue constraints), which are a particular kind of implied

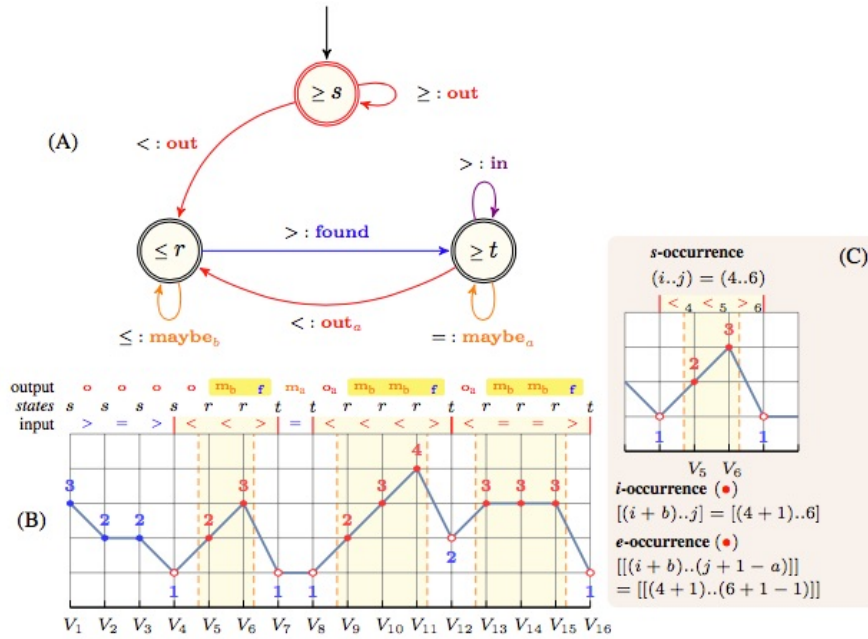


Figure 2. Transducer for the peak pattern and its execution on a sequence

constraints, as well as aggregation bounds (see Figure 5) that can be added to the decomposition of time-series constraints. We evaluate the beneficial propagation impact of the derived implied constraints and bounds, both alone and together. (see [CP conference paper](#))

	$s$	$r$	$t$
$s$	$\phi_g(\vec{C}, \vec{C})$	$\phi_g(\vec{C}, \vec{C})$	$\phi_g(\vec{C}, \vec{C})$
$r$	$\phi_g(\vec{C}, \vec{C})$	$\phi_f(\vec{D}, \vec{D}, \delta_f^i)$	$\phi_f(\vec{C}, \vec{D}, \vec{D}, \delta_f^i)$
$t$	$\phi_g(\vec{C}, \vec{C})$	$\phi_f(\vec{C}, \vec{D}, \vec{D}, \delta_f^i)$	$\phi_g(\vec{C}, \vec{C})$

Figure 3. Parametrised glue matrix for the peak pattern expressed in term of parametrised functions depending on the states pairs between the prefix and the suffix of a sequence

## 7.4. Reformulation of time-series constraint in MIP

A checker for a constraint on a variable sequence can often be compactly specified by an automaton, possibly with accumulators, that consumes the sequence of values taken by the variables; such an automaton can also be used to decompose its specified constraint into a conjunction of logical constraints. The inference achieved by this decomposition in a CP solver can be boosted by automatically generated implied constraints on the accumulators, provided the latter are updated in the automaton transitions by linear expressions. Automata with non-linear accumulator updates can be automatically synthesised for a large family of time-series constraints.

	$s$	$r$	$t$
$s$	$\vec{c} + \overleftarrow{c}$	$\vec{c} + \overleftarrow{c}$	$\vec{c} + \overleftarrow{c}$
$r$	$\vec{c} + \overleftarrow{c}$	1	1
$t$	$\vec{c} + \overleftarrow{c}$	1	$\vec{c} + \overleftarrow{c}$

Figure 4. Concrete glue matrix for the number peak constraint expressed in term of concrete functions depending on the states pairs between the prefix and the suffix of a sequence

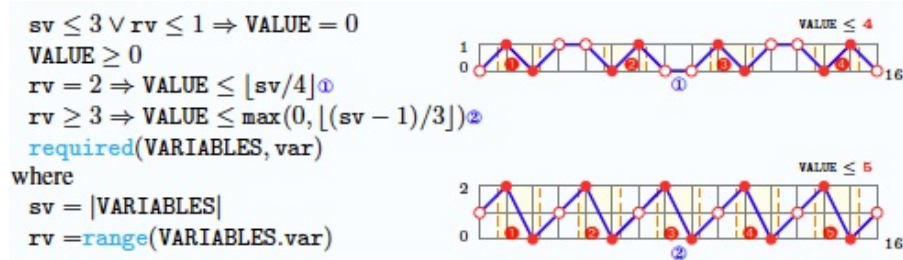


Figure 5. Upper bound on the number of zigzag depending on the domain range being equal to 2 or greater than or equal to 3



In this paper, we describe and evaluate extensions to those techniques. First, we improve the automaton synthesis to generate automata with fewer accumulators. Second, we decompose a constraint specified by an automaton with accumulators into a conjunction of linear inequalities, for use by a MIP solver. Third, we generalise the implied constraint generation to cover the entire family of time-series constraints. The newly synthesised automata for time-series constraints outperform the old ones, for both the CP and MIP decompositions, and the generated implied constraints boost the inference, again for both the CP and MIP decompositions. We evaluate CP and MIP solvers on a prototypical application modelled using time-series constraints. (see [CPAIOR conference paper](#))

## 7.5. Scheduling Constraint for Video Summarisation

Given a sequence of tasks  $T$  subject to precedence constraints between adjacent tasks, and given a set of fixed intervals  $I$ , the TaskIntersection ( $T, I, o, inter$ ) constraint restricts the overall intersection of the tasks of  $T$  with the fixed intervals of  $I$  to be greater than or equal or less than or equal to a given limit  $inter$ . We provide a bound( $Z$ )-consistent cost filtering algorithm wrt the starts and the ends of the tasks for the TaskIntersection constraint and evaluate the constraint on the video summarisation problem. (see [CPAIOR conference paper](#))

## 7.6. A Model Seeker for Learning Constraints Models from Positive Samples

We describe a system which generates finite domain constraint models from positive example solutions (e.g. see Figure 6 giving a season schedule of the Bundesliga), for highly structured problems. The system is based on the global constraint catalog, providing the library of constraints that can be used in modeling, and the Constraint Seeker tool, which finds a ranked list of matching constraints given one or more sample call patterns (e.g. see Figure 7 giving the model learned for the input data of Figure 6). We have tested the modeler with 230 examples, ranging from 4 to 6,500 variables, using between 1 and 7,000 samples. These examples come from a variety of domains, including puzzles, sports-scheduling, packing and placement, and design theory. When comparing against manually specified canonical models for the examples, we achieve a hit rate of 50 percent, processing the complete benchmark set in less than one hour on a laptop. Surprisingly, in many cases the system finds usable candidate lists even when working with a single, positive example. (see [Book chapter of Data Mining and Constraint Programming](#))

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
8	1	14	11	4	7	2	15	12	13	6	9	10	3	18	5	16	17
3	14	17	2	13	6	5	12	9	16	11	18	1	4	15	8	7	10
...																	
18	17	2	1	4	3	6	5	10	9	16	15	14	13	12	11	8	7
13	12	11	14	17	16	15	2	9	6	1	8	7	4	5	18	3	10
...																	

Figure 6. Input data corresponding to a flat sample (a sequence of integer values) giving a one year season schedule of the Bundesliga

## 7.7. Global Constraint Catalog Volume II: Time-Series Constraints

First this report presents a restricted set of 22 finite transducers used to synthesise structural time-series constraints described by means of a multi-layered function composition scheme. Second it provides the

-	Sequence Generator	Projection	Constraint Conjunction
1	scheme(612,34,18,34,1)	id	alldifferent*18
2	scheme(612,34,18,2,2)	id	alldifferent*153
3	scheme(612,34,18,1,18)	id	alldifferent*34
4	scheme(612,34,18,1,18)	absolute_value	symmetric_alldifferent([1..18])*34
5	scheme(612,34,18,17,1)	absolute_value	alldifferent*36
6	repart(612,34,18,34,9)	id	sum_ctr(0)*306
7	repart(612,34,18,34,9)	id	twin*1
8	repart(612,34,18,34,9)	id	elements([i,-i])*1
9	first(9,[1,3,5,7,9,11,13,15,17])	id	strictly_increasing*1
10	vector(612)	id	global_cardinality([-18..-1-17,0-0,1..18-17])*1
11	repart(612,34,18,34,9)	id	sum_powers5_ctr(0)*306
12	repart(612,34,18,34,9)	id	sum_cubes_ctr(0)*306
13	repart(612,34,18,34,3)	sign	global_cardinality([-1-3,0-0,1-3])*102
14	scheme(612,34,18,34,1)	sign	global_cardinality([-1-17,0-0,1-17])*18
15	repart(612,34,18,17,9)	sign	global_cardinality([-1-2,0-0,1-2])*153
16	repart(612,34,18,2,9)	sign	global_cardinality([-1-17,0-0,1-17])*18
17	scheme(612,34,18,1,18)	sign	global_cardinality([-1-9,0-0,1-9])*34
18	repart(612,34,18,34,9)	sign	sum_ctr(0)*306
19	repart(612,34,18,34,9)	sign	twin*1
20	repart(612,34,18,34,9)	absolute_value	twin*1
21	repart(612,34,18,34,9)	sign	elements([i,-i])*1
22	scheme(612,34,18,34,1)	sign	among_seq(3,[-1])*18
23	repart(612,34,18,34,9)	absolute_value	elements([i,i])*1
24	first(9,[1,3,5,7,9,11,13,15,17])	absolute_value	strictly_increasing*1
25	first(6,[1,4,7,10,13,16])	absolute_value	strictly_increasing*1
26	scheme(612,34,18,34,1)	absolute_value	nvalue(17)*18

Figure 7. Model, i.e. conjunction of global constraints, learned from the single flat sample

corresponding synthesised catalogue of structural time-series constraints where each of the 626 constraints is explicitly described in terms of automata with accumulators, see Figure 8 for the synthesised automaton of the sum surf peak constraint. ([arXiv 1609.08925](#))

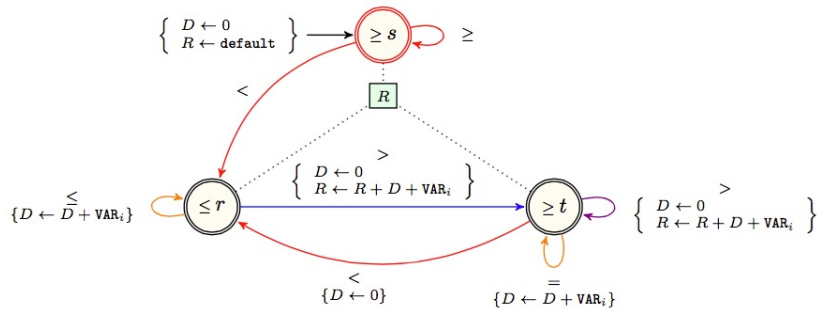


Figure 8. Synthesised automaton with accumulator of the sum surf peak constraint obtained from the transducer of the peak pattern

## 7.8. Probabilistic Model for Binary CSP

This work introduces a probabilistic-based model for binary CSP that provides a fine grained analysis of its internal structure. Assuming that a domain modification could occur in the CSP, it shows how to express, in a predictive way, the probability that a domain value becomes inconsistent, then it express the expectation of the number of arc-inconsistent values in each domain of the constraint network. Thus, it express the expectation of the number of arc-inconsistent values for the whole constraint network. Next, it provides bounds for each of these three probabilistic indicators. Finally, a polytime algorithm, which propagates the probabilistic information, is presented. (see [arXiv 1606.03894](#) or [19])

## 7.9. Estimating parallel runtimes for randomized algorithms in constraint solving

We present a detailed analysis of the scalability and parallelisation of Local Search algorithms for constraint-based and SAT (Boolean satisfiability) solvers. We propose a framework to estimate the parallel performance of a given algorithm by analyzing the runtime behavior of its sequential version. Indeed, by approximating the runtime distribution of the sequential process with statistical methods, the runtime behavior of the parallel process can be predicted by a model based on order statistics. We apply this approach to study the parallel performance of a constraint-based Local Search solver (Adaptive Search), two SAT Local Search solvers (namely Sparrow and CCASAT), and a propagation-based constraint solver (Gecode, with a random labeling heuristic). We compare the performance predicted by our model to actual parallel implementations of those methods using up to 384 processes. We show that the model is accurate and predicts performance close to the empirical data. Moreover, as we study different types of problems, we observe that the experimented solvers exhibit different behaviors and that their runtime distributions can be approximated by two types of distributions: exponential (shifted and non-shifted) and lognormal. Our results show that the proposed framework estimates the runtime of the parallel algorithm with an average discrepancy of 21 percent w.r.t. the empirical data across all the experiments with the maximum allowed number of processors for each technique. (see [Journal of Heuristics](#))

## 7.10. ghost: A Combinatorial Optimization Framework for Real-Time Problems

We presents GHOST, a combinatorial optimization framework that a real-time strategy (RTS) AI developer can use to model and solve any problem encoded as a constraint satisfaction/optimization problem (CSP/COP). We show a way to model three different problems as a CSP/COP, using instances from the RTS game StarCraft as test beds. Each problem belongs to a specific level of abstraction (the target selection as reactive control problem, the wall-in as a tactics problem, and the build order planning as a strategy problem). In our experiments, GHOST shows good results computed within some tens of milliseconds. We also show that GHOST outperforms state-of-the-art constraint solvers, matching them on the resources allocation problem, a common combinatorial optimization problem. (see [IEEE Transactions on Computational Intelligence and AI in games journal](#))

## 7.11. TorchCraft: a Library for Machine Learning Research on Real-Time Strategy Games

We present TorchCraft, a library that enables deep learning research on Real-Time Strategy (RTS) games such as StarCraft: Brood War, by making it easier to control these games from a machine learning framework, here Torch. This white paper argues for using RTS games as a benchmark for AI research, and describes the design and components of TorchCraft. (see [arXiv 1611.00625](#))

## 7.12. POSL: A Parallel-Oriented metaheuristic-based Solver Language

For a couple of years, all processors in modern machines are multi-core. Massively parallel architectures, so far reserved for super-computers, become now available to a broad public through hardware like the Xeon Phi or GPU cards. This architecture strategy has been commonly adopted by processor manufacturers, allowing them to stick with Moore's law. However, this new architecture implies new ways to design and implement algorithms to exploit its full potential. This is in particular true for constraint-based solvers dealing with combinatorial optimization problems. Here we propose a Parallel-Oriented Solver Language (POSL, pronounced "puzzle"), a new framework to build interconnected meta-heuristic based solvers working in parallel. The novelty of this approach lies in looking at solver as a set of components with specific goals, written in a parallel-oriented language based on operators. A major feature in POSL is the possibility to share not only information, but also behaviors, allowing solver modifications during runtime. Our framework has been designed to easily build constraint-based solvers and reduce the developing effort in the context of parallel architecture. POSL's main advantage is to allow solver designers to quickly test different heuristics and parallel communication strategies to solve combinatorial optimization problems, usually time-consuming and very complex technically, requiring a lot of engineering.

## 7.13. Towards Automated Strategies in Satisfiability Modulo Theory

SMT solvers include many heuristic components in order to ease the theorem proving process for different logics and problems. Handling these heuristics is a non-trivial task requiring specific knowledge of many theories that even a SMT solver developer may be unaware of. This is the first barrier to break in order to allow end-users to control heuristics aspects of any SMT solver and to successfully build a strategy for their own purposes. We present a first attempt for generating an automatic selection of heuristics in order to improve SMT solver efficiency and to allow end-users to take better advantage of solvers when unknown problems are faced. Evidence of improvement is shown and the basis for future works with evolutionary and/or learning-based algorithms are raised (see [Genetic Programming conference paper](#)).

## 7.14. Using CP for the Urban Transit Crew Rescheduling Problem

Scheduling urban and trans-urban transportation is an important issue for industrial societies. The Urban Transit Crew Scheduling Problem is one of the most important optimization problem related to this issue. It mainly relies on scheduling bus drivers workday respecting both collective agreements (see Figure 9 for an example of regulation rule) and the bus schedule needs. If this problem has been intensively studied from a tactical point of view, its operational aspect has been neglected while the problem becomes more and more complex and more and more prone to disruptions. In this way, this paper presents how the constraint programming technologies are able to recover the tactical plans at the operational level in order to efficiently help in answering regulation needs after disruptions (see [CP conference paper](#)).

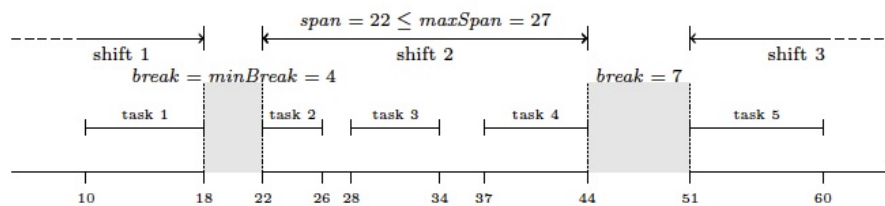


Figure 9. Illustration of typical regulation rule in the Labcom project; Shift 2 of an employee is composed of tasks 2, 3, and 4. It is between shifts 1 and 3 of the same employee. A break of duration 4 is scheduled between task 1 of shift 1 and task 2 of shift 2, because the gap between these tasks is at least a break of 4. Similarly, a break of duration 7 is scheduled after task 4 of shift 2 and task 5 of shift 3. No other breaks can be scheduled between the tasks of shift 2 because of the minimum break duration. The span of shift 2 is 22 and does not exceed 27: it is composed of tasks 2, 3, and 4, as well as of the two gaps between these tasks.

## 7.15. Traveling salesman and the tree: the importance of search in CP

The traveling salesman problem (TSP) is a challenging optimization problem for CP and OR that has many industrial applications. Its generalization to the degree constrained minimum spanning tree problem (DCMSTP) is being intensively studied by the OR community. In particular, classical solution techniques for the TSP are being progressively generalized to the DCMSTP. Recent work on cost-based relaxations has improved CP models for the TSP. However, CP search strategies have not yet been widely investigated for these problems. The contributions of this paper are twofold. We first introduce a natural generalization of the weighted cycle constraint (WCC) to the DCMSTP. We then provide an extensive empirical evaluation of various search strategies. In particular, we show that significant improvement can be achieved via our graph interpretation of the state-of-the-art Last Conflict heuristic. (see [Constraints journal](#))

## 7.16. Event Selection Rules to Compute Explanations

Explanations have been introduced in the previous century. Their interest in reducing the search space is no longer questioned. Yet, their efficient implementation into CSP solver is still a challenge. In this paper, we introduce ESeR, an Event Selection Rules algorithm that filters events generated during propagation. This dynamic selection enables an efficient computation of explanations for intelligent backtracking algorithms. We show the effectiveness of our approach on the instances of the last three MiniZinc challenges. (see [arXiv 1608.08015](#) or [20])

## 7.17. Towards energy-proportional Clouds partially powered by renewable energy

With the emergence of the Future Internet and the dawning of new IT models such as cloud computing, the usage of data centers (DC), and consequently their power consumption, increase dramatically. Besides the ecological impact, the energy consumption is a predominant criterion for DC providers since it determines the daily cost of their infrastructure. As a consequence, power management becomes one of the main challenges for DC infrastructures and more generally for large-scale-distributed systems. In this paper, we present the EpoCloud prototype, from hardware to middleware layers. This prototype aims at optimizing the energy consumption of mono-site Cloud DCs connected to the regular electrical grid and to renewable-energy sources (see [Journal of Computing](#)).

# 8. Bilateral Contracts and Grants with Industry

## 8.1. Bilateral Contracts with Industry

### 8.1.1. Labcom TransOp

**Participants:** Charles Prud'Homme, Xavier Lorca.

Title: TransOp.

Duration: 2014-2016.

Type: **ongoing project**.

Others partners: [Eurodécision](#).

The goal of the project is to handle robustness in the context of industrial timetabling problems with constraint programming using [CHOCO](#). The project is managed by [Xavier Lorca](#).

## 8.2. Bilateral Grants with Industry

### 8.2.1. Gaspard Monge

**Participants:** Nicolas Beldiceanu, Helmut Simonis.

Title: Gaspard Monge 3.

Duration: 2016.

Type: **ongoing project**.

Others partners: EDF.

Within the context of the Gaspard Monge call program for Optimisation and Operation Research, we work with [EDF](#) on the research initiative on *Optimization and Energy*. The goal of the project (continuation of last years projects) is to provide a systematic reformulation of time-series constraints in term of linear constraints that can be used in a MIP solver.

# 9. Partnerships and Cooperations

## 9.1. Regional Initiatives

### 9.1.1. EPOC

With the emergence of the Future Internet and the dawning of new IT models such as cloud computing, the usage of data centers (DC), and consequently their power consumption, increase dramatically. Besides the ecological impact, the energy consumption is a predominant criteria for DC providers since it determines the daily cost of their infrastructure. As a consequence, power management becomes one of the main challenges for DC infrastructures and more generally for large-scale distributed systems. In this paper, the EPOC project which focuses on optimising the energy consumption of mono-site DCs connected to the regular electrical grid and to renewable energy sources.

### 9.1.2. *SmartCat*

**Participants:** Eric Monfroy, Charlotte Truchet.

Title: Online optimization for chemical reactions.

Others partners: **CEISAM**.

The SmartCat project, started in 2015 on regional fundings, aims at developing an intelligent automatised tool for online chemistry. Contrarily to the traditional batch chemistry, where reactants are mixed in a glass, online chemistry consists in having a flow of reactants in a tube, possibly passing through ovens are pressure control mechanisms. This way, the reaction happens continuously and it can produce much more products within a system of reasonable size. SmartCat integrates a controller for which intelligent tools need to be developed. These tools will analyse the product of the reaction and adapt the conditions (stoichiometry, pressure, temperature, catalysis) in order to optimise the yield. TASC contributes to this project by developing these methods, based on local search techniques.

### 9.1.3. *Atlantisc 1*

**Participant:** Florian Richoux.

Title: Atlantisc project about deep learning for games.

Duration: 2016.

Topic: deep learning for games.

### 9.1.4. *Atlantisc 2*

**Participant:** Charles Prud Homme.

Title: CoMe4ACloud.

Duration: 2016.

Topic: CoMe4ACloud is an Atlantisc2020 funded project whose objective is to provide an end-to-end solution for autonomic Cloud services. To that end, we rely on techniques of Constraint Programming so as a decision-making tool and Model-driven Engineering to ease the automatic generation of the so-called autonomic managers as well as their communication with the managed system (see **Constraints and Model Engineering for Autonomic Clouds**). The project is led by ASCOLA research team and involves also AtlanModels and TASC.

## 9.2. National Initiatives

### 9.2.1. *ANR NetWMS2*

**Participants:** Gilles Chabert, Ignacio Salas Donoso, Nicolas Beldiceanu.

Title: Networked Warehouse Management Systems 2: packing with complex shapes.

Duration: 2011-2014.

Type: cosinus research program.

Budget: 189909 Euros.

Others partners: **KLS Optim** and **CONTRAINTEs** (Inria Rocquencourt).

This project builds on the former European FP6 **Net-WMS** Strep project that has shown that constraint-based optimisation techniques can considerably improve industrial practice for box packing problems, while identifying hard instances that cannot be solved optimally, especially in industrial 3D packing problems with rotations, the needs for dealing with more complex shapes (e.g. wheels, silencers) involving continuous values. This project aims at generalizing the geometric kernel *geost* for handling non-overlapping constraints for complex two and three dimensional curved shapes as well as domain specific heuristics. This will be done within the continuous solver **IBEX**, where discrete variables will be added for handling polymorphism (i.e., the fact that an object can take one shape out of a finite set of given shapes). A filtering algorithm has been devised in the case of objects described by nonlinear inequalities and is now under testing with the Ibex library. This work has been presented in a workshop on interval methods & geometry in **ENSTA Bretagne**.

## 9.3. European Initiatives

### 9.3.1. FP7 & H2020 Projects

Within the context of the **First Future and Emerging Technologies (FET) Proactive projects under Horizon 2020 Framework Programme** the **GRACeFUL** project started this year. From an application point of view the project develops scalable rapid assessment tools for collective policy making in global systems, and test these on climate-resilient urban design. From a technical point of view it provides domain specific languages that are embedded in functional programming and constraint programming languages. Within the project TASC is responsible for the constraint part. To interact with policy makers it uses some qualitative network model (see Figure 10) embedded with constraint programming models that also capture dependency between potential actions as well as costs.

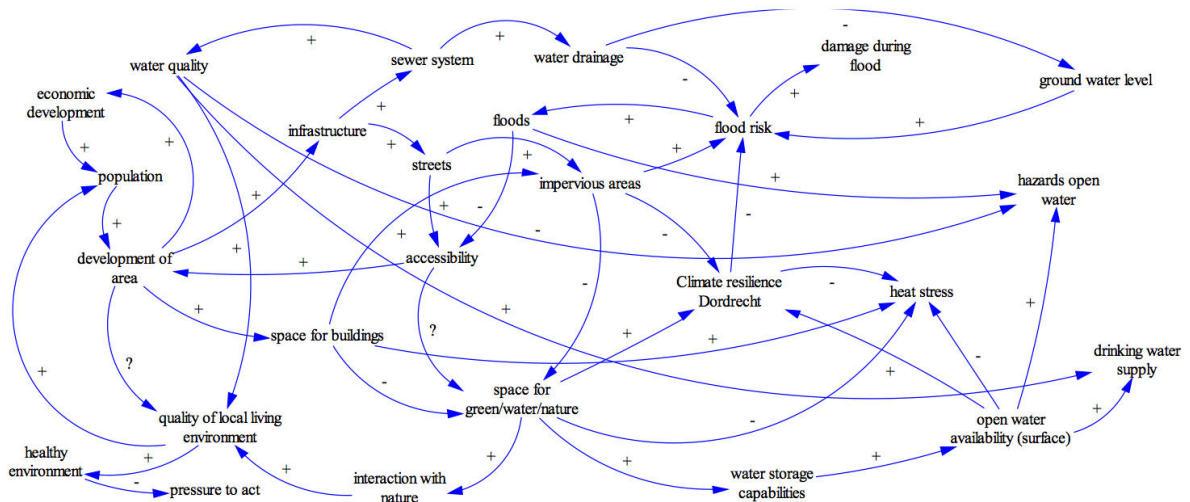


Figure 10. Illustration of some qualitative network capturing causality in the context of flooding prevention

## 9.4. International Initiatives

### 9.4.1. Inria Associate Teams Not Involved in an Inria International Labs

#### 9.4.1.1. TASC MELB

Title: Synergy between Filtering and Explanations for Scheduling and Placement Constraints

International Partner (Institution - Laboratory - Researcher):

NICTA (Australia) - Optimisation Research Group (Optimisation) - Pascal van Hentenryck

Start year: 2014

See also: <http://www.normalesup.org/~truchet/TASC MELB.html>

In the context of Constraint Programming and SAT the project addresses the synergy between filtering (removing values from variables) and explanations (explaining why values were removed in term of clauses) in order to handle in a more efficient way correlated resource scheduling and placement constraints. It combines the strong point of Constraint Programming, namely removing



value that leads to infeasibility, with the strong point of SAT, namely taking advantage from past failure in order to quickly identify infeasible sub-problems. In 2016 we got the following new result *using rewriting for synthesising filtering algorithm for the Allen constraint*: For all 8192 combinations of Allen's 13 relations between one task with origin  $o_i$  and fixed length  $l_i$  and another task with origin  $o_j$  and fixed length  $l_j$ , we give a formula evaluating to a set of integers which are infeasible for a task origin for the given combination. Such forbidden regions are useful e.g. in a range-consistency maintaining propagator for an Allen constraint in finite domain constraint programming. No visit to Melbourne was done this year because of VISA problem. Consequently we also did remotely (i.e. from Nantes) the following result: the availability of the time-series constraints of the time-series constraint catalog available in the MiniZinc modelling language (and consequently made them accessible to solvers like Choco or Cplex).

## 9.5. International Research Visitors

### 9.5.1. Visits of International Scientists

- A visit regarding time-series constraints of [Andreina Francisco Rodriguez Helmut Simonis](#), [Pierre Flener](#) and [Justin Pearson](#) in Nantes in May.
- A visit regarding time-series constraints of [Helmut Simonis](#), in July in May.

### 9.5.2. Visits to International Teams

- Two visits of E. Arafailova regarding time-series constraints in Cork (March 2016) and in Uppsala (April 2016)
- Three visits of N. Beldiceanu regarding time-series constraint in Cork (June 2016) and in Uppsala (February 2016, August 2016)

## 10. Dissemination

### 10.1. Promoting Scientific Activities

#### 10.1.1. Leadership within the Scientific Community

Charlotte Truchet was elected in the ACP committee.

#### 10.1.2. Research Administration

Preparation by the whole team of the two evaluations that respectively took place in January 2016 (hceres evaluation) and in March 2016 (inria evaluation).

### 10.2. Teaching - Supervision - Juries

#### 10.2.1. Supervision

PhD: [Ignacio Salas Donoso](#), Packing curved objects with interval methods, Started in May 2013, PhD defense April 5 2016 with committee Luc Jaulin, Gilles Trombettoni, François Fages, see [PhD thesis](#), [Gilles Chabert](#) and [Nicolas Beldiceanu](#).

PhD in progress : [Gilles Madi Wamba](#), Mixing constraint programming and behavioural models to manage energy consumption in data centre, October 2014, [Nicolas Beldiceanu](#) and [Didier Lime](#).

PhD in progress : [Alejandro Reyes Amaro](#), Toward autonomous parallel algorithms for constraint-based problems, October 2014, [Eric Monfroy](#) and [Florian Richoux](#).

PhD in progress : [Anicet Bart](#), Solving mixed constraints, application to the management of mobile sensors, October 2014, [Eric Monfroy](#) and [Charlotte Truchet](#).

PhD in progress : **Ekaterina Arafailova**, Functional constraints, September 2015, **Nicolas Beldiceanu** and **Rémi Douence**.

PhD in progress : **Nicolas Galvez**, Hybrid Algorithms for Search Based Software Engineering, December 2014, **Eric Monfroy** with **Frédéric Saubion** from Angers University and **C. Castro** from UTFSM Valparaiso, Chili.

### 10.3. Popularization

- Maintenance of the global constraint catalogue.
- Illustrations of the volume II of the global constraint catalogue: 2000 figures.

## 11. Bibliography

### Publications of the year

#### Articles in International Peer-Reviewed Journals

- [1] N. BELDICEANU, M. CARLSSON, R. DOUENCE, H. SIMONIS. *Using finite transducers for describing and synthesising structural time-series constraints*, in "Constraints", January 2016, vol. 21, n<sup>o</sup> 1 [DOI : 10.1007/s10601-015-9200-3], <https://hal.inria.fr/hal-01370322>
- [2] N. BELDICEANU, B. DUMAS FERIS, P. GRAVEY, S. HASAN, C. JARD, T. LEDOUX, Y. LI, D. LIME, G. MADI-WAMBA, J.-M. MENAUD, P. MOREL, M. MORVAN, M.-L. MOULINARD, A.-C. ORGERIE, J.-L. PAZAT, O. H. ROUX, A. SHARAIHA. *Towards energy-proportional Clouds partially powered by renewable energy*, in "Computing", January 2017, vol. 99, n<sup>o</sup> 1, 20 p. [DOI : 10.1007/s00607-016-0503-z], <https://hal.inria.fr/hal-01340318>
- [3] J.-G. FAGES, X. LORCA, L.-M. ROUSSEAU. *The salesman and the tree: the importance of search in CP*, in "Constraints", 2016, vol. 21, pp. 145 - 162 [DOI : 10.1007/s10601-014-9178-2], <https://hal.archives-ouvertes.fr/hal-01436214>
- [4] J.-N. MONETTE, N. BELDICEANU, P. FLENER, J. PEARSON. *A Parametric Propagator for Pairs of Sum Constraints with a Discrete Convexity Property*, in "Artificial Intelligence", 2016, vol. 241, pp. 170 - 190 [DOI : 10.1016/J.ARTINT.2016.08.006], <https://hal.inria.fr/hal-01374714>
- [5] F. RICHOUX, A. URIARTE, J.-F. BAFFIER. *GHOST: A Combinatorial Optimization Framework for Real-Time Problems*, in "IEEE Transactions on Computational Intelligence and AI in games", 2016, vol. 8, n<sup>o</sup> 4, pp. 377-388 [DOI : 10.1109/TCIAIG.2016.2573199], <https://hal.archives-ouvertes.fr/hal-01436124>

#### International Conferences with Proceedings

- [6] E. ARAFAILOVA, N. BELDICEANU, R. DOUENCE, P. FLENER, M. A. FRANCISCO RODRÍGUEZ, J. PEARSON, H. SIMONIS. *Time-Series Constraints: Improvements and Application in CP and MIP Contexts*, in "CPAIOR 2016 - 13th International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming", Banff, Canada, C.-G. QUIMPER (editor), Lecture Notes in Computer Science, Springer, May 2016, vol. 9676, pp. 18-34 [DOI : 10.1007/978-3-319-33954-2], <https://hal.inria.fr/hal-01355262>

- [7] X. LORCA, C. PRUD'HOMME, A. QUESTEL, B. ROTTEMBOURG. *Using Constraint Programming for the Urban Transit Crew Rescheduling Problem*, in "Principles and Practice of Constraint Programming", Toulouse, France, September 2016, vol. 9892, pp. 636 - 649 [DOI : 10.1007/978-3-319-44953-1\_40], <https://hal.archives-ouvertes.fr/hal-01436288>

### **National Conferences with Proceedings**

- [8] A. REYES AMARO, E. MONFROY, F. RICHOUX. *Étude de stratégies parallèles de coopération avec POSL*, in "Douzièmes Journées Francophones de Programmation par Contraintes (JFPC)", Montpellier, France, June 2016, <https://hal.archives-ouvertes.fr/hal-01436130>

### **Conferences without Proceedings**

- [9] A. BART, C. TRUCHET, E. MONFROY. *Contraintes sur des flux appliquées à la vérification de programmes audio*, in "Onzièmes Journées Francophones de Programmation par Contraintes", Bordeaux, France, June 2016, <https://hal.archives-ouvertes.fr/hal-01234179>

- [10] G. MADI WAMBA, N. BELDICEANU. *The TaskIntersection Constraint*, in "Integration of AI and OR techniques in constraint programming", Banff, Canada, Integration of AI and OR techniques in constraint programming, Springer, May 2016, vol. 9676, pp. 246-261 [DOI : 10.1007/978-3-319-33954-2\_18], <https://hal.archives-ouvertes.fr/hal-01436044>

### **Scientific Books (or Scientific Book chapters)**

- [11] N. BELDICEANU, H. SIMONIS. *ModelSeeker: Extracting Global Constraint Models from Positive Examples*, in "Data Mining and Constraint Programming", Data Mining and Constraint Programming, Springer, December 2016, vol. 10101, pp. 77-95 [DOI : 10.1007/978-3-319-50137-6\_4], <https://hal.archives-ouvertes.fr/hal-01436062>

- [12] D. CHURCHILL, M. PREUSS, F. RICHOUX, G. SYNNAEVE, A. URIARTE, S. ONTAÑÓN, M. CERTICKY. *StarCraft Bots and Competitions*, in "Encyclopedia of Computer Graphics and Games", N. LEE (editor), Springer International Publishing, 2016, pp. 1-18 [DOI : 10.1007/978-3-319-08234-9\_18-1], <https://hal.archives-ouvertes.fr/hal-01436110>

- [13] A. REYES AMARO, E. MONFROY, F. RICHOUX. *POSL: A Parallel-Oriented metaheuristic-based Solver Language*, in "Springer Recent Developments of Metaheuristics", Springer, 2016, <https://hal.archives-ouvertes.fr/hal-01436119>

### **Scientific Popularization**

- [14] A. AGGOUN, N. BELDICEANU, G. CHABERT, F. FAGES. *Packing with Complex Shapes*, in "ERCIM News", 2016, n° 105, 32 p., <https://hal.archives-ouvertes.fr/hal-01378471>

### **Other Publications**

- [15] E. ARAFAILOVA, N. BELDICEANU, M. CARLSSON, P. FLENER, M. A. FRANCISCO RODRÍGUEZ, J. PEARSON, H. SIMONIS. *Systematic Derivation of Bounds and Glue Constraints for Time-Series Constraints*, August 2016, slides corresponding to this paper [DOI : 10.1007/978-3-319-44953-1\_2], <https://hal.inria.fr/hal-01370317>

- [16] R. DOUENCE, P. FLENER, M. A. F. RODRÍGUEZ, J. PEARSON, H. SIMONIS, M. A. FRANCISCO RODRÍGUEZ, M. CARLSSON, N. BELDICEANU. *Global Constraint Catalog, Volume II, Time-Series Constraints*, September 2016, 2709 p. , Synthesized catalogue (text, figures, code) of time-series constraints, <https://hal.inria.fr/hal-01374721>
- [17] A. REYES-AMARO, E. MONFROY, F. RICHOUX. *Supplementary document for Studying parallel communication strategies with POSL*, July 2016, working paper or preprint, <https://hal.archives-ouvertes.fr/hal-01345775>
- [18] G. SYNNAEVE, N. NANTAS, A. ALEX, S. CHINTALA, L. TIMOTHÉE, L. ZEMING, F. RICHOUX, N. USUNIER. *TorchCraft: a Library for Machine Learning Research on Real-Time Strategy Games*, January 2017, working paper or preprint, <https://hal.archives-ouvertes.fr/hal-01436134>

## References in notes

- [19] A. BALAFREJ, X. LORCA, C. TRUCHET. *A Probabilistic-Based Model for Binary CSP*, in "CoRR", 2016, vol. abs/1606.03894, <http://arxiv.org/abs/1606.03894>
- [20] C. PRUD'HOMME, X. LORCA, N. JUSSIEN. *Event Selection Rules to Compute Explanations*, in "CoRR", 2016, vol. abs/1608.08015, <http://arxiv.org/abs/1608.08015>