

R&D Project Report

Franco-Japanese Research Collaboration on Constraint Programming

Frédéric Benhamou¹, Martine Ceberio², Philippe Codognet³, Hiroshi Hosobe⁴,
Christophe Jermann⁵, Ken Satoh⁶, and Kazunori Ueda⁷

^{1,5}University of Nantes

² University of Texas at El Paso

³ Embassy of France in Japan / University of Paris 6

^{4,6}National Institute of Informatics / The Graduate University for Advanced Studies

⁷ Waseda University

Abstract

Constraint programming is an emergent technology that allows modeling and solving various problems in many areas such as artificial intelligence, computer programming, computer-aided design, computer graphics, and user interfaces. In this report, we provide recent activities of research collaboration on constraint programming conducted by the authors and other researchers in France and Japan. First, we outline our joint research projects on constraint programming, and then present the backgrounds, goals, and approaches of several research topics treated in the projects. Second, we describe the two Franco-Japanese Workshops on Constraint Programming (FJCP), which we organized in Japan in October 2004 and in France in November 2005. We conclude with future prospects for collaboration between French and Japanese researchers in this area.

Keywords

constraint programming, constraint logic programming, speculative computation, multi-agent systems, constraint hierarchies, global optimization

1 Introduction

Constraint programming is an emergent technology that allows modeling and solving various problems in many areas such as artificial intelligence, computer programming, computer-aided design,

computer graphics, and user interfaces. It allows declarative specification of problems with constraints that express relationships among objects, and also it enables automatic maintenance of solutions to the specified problems by using constraint solvers.

Constraint programming has its primary origin in logic programming. In the middle 1980's, constraint logic programming was developed as a generalization of logic programming by integrating constraints [3]. In that development, France and Japan played important roles. In France, signifi-

¹)Frederic.Benhamou@lina.univ-nantes.fr

²)mceberio@cs.utep.edu

³)Philippe.Codognet@lip6.fr

⁴)hosobe@nii.ac.jp

⁵)Christophe.Jermann@lina.univ-nantes.fr

⁶)ksatoh@nii.ac.jp

⁷)ueda@ueda.info.waseda.ac.jp

cant achievements on constraint logic programming languages such as Prolog III [5] were made. In Japan, a national project called the Fifth Generation Computer Systems founded the Institute for New Generation Computer Technology (ICOT) in 1982, and conducted research on various aspects of logic programming including its constraint extension.

In this report, we provide recent activities of research collaboration on constraint programming conducted by the authors and other researchers in France and Japan. We believe that this is a unique collaboration between France and Japan in this research area. In fact, not only are we currently working in this area, but also some of us were involved in achievements in constraint logic programming around 1990 in these countries.

The rest of this report is organized as follows. In Section 2, we outline our joint research projects on constraint programming, and then present the backgrounds, goals, and approaches of several research topics treated in the projects. In Section 3, we describe the two Franco-Japanese Workshops on Constraint Programming (FJCP), which we organized in Japan in October 2004 and in France in November 2005. In Section 4, we conclude with future prospects for collaboration between French and Japanese researchers in this area.

2 Joint Research

This section first overviews the history of our joint research, and then describes the outlines of our three main research topics.

2.1 Overview

Our joint research dates back to the year 2002. From June to August 2002, Codognet served as a Visiting Professor at the National Institute of Informatics (NII). He collaborated with Satoh and Hosobe in studying the use of speculative computation in constraint processing. (See Subsection 2.2 for details.)

Since 2004, we have been conducting several joint research projects. The most general project is called SCooP, which stands for Soft and Continuous Constraint Programming. This project is being conducted under the Memorandum of Understand-

ing (MOU) that was signed between Laboratoire d'Informatique de Nantes Atlantique (LINA) and NII. This project treats constraint programming related to soft constraints and continuous constraints in a wide sense, which covers a broad range of our research interests. Therefore, we regard most research collaboration between LINA and NII as activities within this SCooP project. NII provides researchers involved in the MOU-related projects with opportunities to visit their counterpart organizations, which allowed some of the authors to visit LINA or NII to progress the SCooP project.

Our first project that treated a more concrete research topic was titled "Continuous Soft Constraint Programming for Graphical Interface Applications." The aim of this project was to develop new methods that use global optimization and its related techniques to handle the framework of soft constraints called constraint hierarchies. (See Subsections 2.3 and 2.4.) Benhamou and Satoh applied to the Japan-France Integrated Action Program (SAKURA), conducted by the French Leading Agency for International Mobility (Égide) and the Japan Society for the Promotion of Science (JSPS), and this project was accepted and funded by the program from 2004 to 2005. (Therefore, we often simply call this project "the SAKURA project.") The fund mainly supported our short-term trips from France to Japan or vice versa.

In 2004, we were also supported by the NII joint research project "A Study on Speculative Constraint Processing in Multi-Agent Systems," which allowed Ceberio to stay at NII in October 2004 for a month. She joined with Satoh and Hosobe in extending our previous method of speculative constraint processing. (See Subsection 2.2.)

The University of Nantes provided Hosobe with an opportunity to serve as "Enseignant Invité" (Invited Professor) at LINA for two months from early May 2005. He collaborated with Benhamou and Jermann in studying research topics related to the SCooP and SAKURA projects.

We were also supported by NII's Research Center for Testbeds and Prototyping (RCTP) from early 2004 to March 2006. It was complementary to the other support, in that RCTP supported us mainly in preparing software and hardware that are needed to conduct our joint research.

The rest of this section is focused on three specific topics treated in our joint research.

2.2 Speculative Constraint Processing

Multi-agent systems allow problem solving in which distributed multiple computational agents cooperate by communicating with each other. In most current research on multi-agent systems, communication between agents is usually assumed to be guaranteed, and typically, agents that have sent queries to other agents need to stop computation until receiving the corresponding replies from the asked agents. However, such an assumption is not always valid in environments such as the Internet, where communication is not guaranteed. It might happen, for example, that a message is lost during communication between agents, or that a reply is delayed quite long due to extremely slow communication. In addition, even if communication is guaranteed, a situation similar to unguaranteed communication might occur. For example, an agent processing a received query and trying to ask its owner about the query might stop the processing if the owner is away at that time.

To handle such situations in multi-agent systems, Satoh et al. previously proposed a method of speculative computation based on abduction [9]. In this method, which is formalized as an extension of logic programming, even if an agent needs to ask another agent about some information, it can speculatively continue computation by using a default hypothesis about the necessary information. The method takes advantage of this speculative computation by allowing the asking agent to exploit the result of the speculative computation when it receives the corresponding reply. This is done by checking the consistency of the reply with the default hypothesis: if the reply is consistent, the agent continues the current computation; otherwise, the agent performs alternative computation that corresponds to the reply.

However, the previous method is limited to master-slave multi-agent systems with only yes/no queries. To tackle the limitation of such queries, we developed a method of speculative constraint processing [8] by extending the previous method to introduce constraints. In this extended method, each query is associated with a constraint that can express more general information than yes/no. It allows an agent to ask another agent about a more complex query at one time, which enables more

efficient communication than in the case of only yes/no queries. To handle such complex queries, we extended the previous method to handle an additional case: that is, the reply does not entail but is consistent with the default hypothesis. In this case, the extended method continues the computation using the default, and simultaneously starts alternative computation as well. For this work [8], we received the Best Paper Award at the Sixth Pacific Rim International Workshop on Multi-Agents.

Furthermore, we tackled another problem with the previous method, that is, its limitation to the master-slave structure of multi-agent systems. As the first step for this purpose, we incorporated belief revision into speculative constraint processing [4]. In the previous method of speculative constraint processing, slave agents were assumed to send at most one final answer. However, this assumption is too strong when we consider more general multi-agent systems, since agents in such systems might want to revise previous answers or to add new answers. (In fact, addition of new answers naturally occurs in computational agents based on logic programming, because they might derive such new answers from alternative computation paths.) Therefore, we developed another method of speculative constraint processing by extending the previous one to handle revised answers and added answers. This extension also gives additional benefits: that is, a slave agent can return to the master agent a tentative reply that has not been confirmed, and also can return to the master agent a partial reply even when it has not yet finished all computation. Furthermore, even in such a case, the master agent can carry out computation that exploits the results of speculative computation. Therefore, the master agent can perform computation more speculatively.

As for future work, we will cover more general forms of multi-agent systems, where every agent can perform speculative computation. To handle a more general multi-agent system, we need to guarantee the appropriate computation of the overall system by additionally considering communication paths among agents.

2.3 Global Optimization Approach to Constraint Hierarchies

Global optimization [7] is the discipline of solving constrained or unconstrained optimization prob-

lems with the goal of achieving the global minimum of the problem. An optimization problem is composed of an objective function (a mathematical expression, often nonlinear) along with a set of constraints (possibly empty, often nonlinear) that restricts the possible assignments for the variables which occur in the objective function. Solving an optimization problem consists of finding an assignment of the variables minimizing the objective function. Such an assignment is a global minimum if no other assignment yield a smaller value of the objective function while satisfying the constraints.

A general framework for translating soft constraint satisfaction problems (CSPs) into global optimization problems was proposed in the first year of the SAKURA project. Basically, it associates a penalty function to each constraint, and a set of aggregators that combine the penalty functions into a global penalty. The *best* solution to the soft CSP corresponds to the assignment which minimizes the global penalty; this assignment can be achieved using global optimization techniques. The interest of this framework is that it can achieve solutions of soft CSPs corresponding to different soft solution criteria by changing the penalty functions and the aggregators.

However, it appeared that modeling exactly constraint hierarchies within this framework is not possible; indeed, the framework produces a single optimization problem while a constraint hierarchy corresponds to a sequence of optimization problems, one for each level in the hierarchy.

In the second half of the SAKURA project, two specific global optimization approaches to constraint hierarchies were studied: the lexicographic optimization and the refining method.

A lexicographic optimization problem is an optimization problem whose objective is an ordered vector of functions $[f_1, \dots, f_n]$. An assignment θ of the variables is a global optimum if any other assignment θ' such that $f_k(\theta) > f_k(\theta')$ also verifies $f_i(\theta) < f_i(\theta')$ for an $i < k$, i.e., the first functions in the objective are always preferred to the other ones. A constraint hierarchy solution according to any global comparator is a global optimum of a lexicographic optimization problem whose objective is the penalty of the different hierarchy levels.

Achieving the exact theoretical solution to a lexicographic optimization problem is difficult in practice, due to the approximation of real numbers

by floating points in computers. Hence, we have defined a new notion of interval global optimum, achievable by interval global optimizers. However, the interval global optimum is in general not guaranteed to include the real global optimum. We are still working on this theoretical aspects.

The refining method, already used in the constraint hierarchy community [10], is a way to obtain the solutions of a constraint hierarchy corresponding to a global comparator. It consists of finding, at each iteration, the assignments that minimize current hierarchy level penalty function; these assignments are searched in the solution set resulting from previous level computation. Hence, at the first iteration the retained assignments are the ones that minimize f_1 , among which only those that minimize f_2 are retained at the second iteration, and so on. This method transforms a constraint hierarchy into a sequence of optimization problems. It may be a way to overcome the theoretical difficulties of interval lexicographic optimization, but the problem of storing the solution sets between each iteration introduces some practical difficulties also.

In the near future, we expect to propose a first practical implementation of a global optimization approach to constraint hierarchies with well-defined theoretical properties.

2.4 Graph-Based Approach to Constraint Hierarchies

The locally-predicate-better (LPB) solutions to a constraint hierarchy basically satisfy maximal consistent subsets of constraints in each hierarchy level. In order to compute those subsets, a maximum-matching based algorithm, called **GR** in this paper, was proposed [6]. It relies on the assumption that each constraint can fix exactly one variable, hence constraints can be matched with variables and inconsistencies appear as matching conflicts, i.e. when several constraints compete in matching the same variable. In such a case, to satisfy the constraint hierarchy definitions, the constraints with highest priority should be matched first. The **GR** algorithm is the most general method proposed in the constraint hierarchy community to achieve LPB solutions.

However, several similar graph-based or flow-based algorithm have been proposed in other communities: in the constraint programming commu-

nity, a maximum-matching based algorithm decomposes a CSP into its over-, well- and under-constrained parts. Well- and under-constrained parts do not require the use of priorities since no constraints conflicts occur in these parts; in the geometric constraint community, flow-based algorithms are used to identify well- or over-constrained subsets. One interest of these algorithms is that they can handle multi-output constraints, i.e. constraints that fix several variables. For instance, the equality between two points in 2D fixes two variables simultaneously, the x and y coordinates of the point.

During the SAKURA project, we have studied the applicability of these graph-based and flow-based methods to constraint hierarchies. It has allowed us to design a novel flow-based algorithm that subsumes algorithm GR since it imposes no restrictions on the constraints in the constraint hierarchy. We are now devising the theoretical properties of this algorithm and implementing a prototype version.

In the near future, we will propose a new definition of constraint hierarchy solutions that better fits the users' expectations in 3D graphic environments. Intuitively, this definition combines local and global comparators in order to satisfy exactly as many constraints as possible, then optimize the violation of the remaining ones. This new definition will be achievable using a graph-based algorithm followed by an optimization process. Applications are expected in the field of virtual reality.

3 FJCP Workshops

This section describes the two Franco-Japanese Workshops on Constraint Programming, which we organized in Japan in 2004 and in France in 2005.

3.1 First Workshop

Late in 2003, the Embassy of France in Japan proposed that we should organize a workshop to enhance collaboration between France and Japan on constraint programming. Accepting that proposal, we set up an organizing committee that consisted of Benhamou, Codognet, Hosobe, Jermann, Satoh and Ueda.

The workshop was realized as the 1st Franco-Japanese Workshop on Constraint Programming (FJCP 2004) [1], which took place at NII in Tokyo, from October 25th to 27th, 2004. It was sponsored by the Embassy of France in Japan as well as LINA, and also was supported by the NII joint research project led by Ueda.

The workshop gathered approximately 30 participants from France and Japan. The program was composed of 24 technical talks, covering both theoretical and practical aspects of constraint programming as well as several applications of constraint technology, in addition to an invited talk given by Krzysztof R. Apt on a rule-based approach to constraint programming.

3.2 Second Workshop

Following the success of FJCP 2004, the 2nd Franco-Japanese Workshop on Constraint Programming (FJCP 2005) [2] occurred at the Port aux Rocs hotel, in Le Croisic, a charming village on the Atlantic coast close to Nantes, from November 14th to 16th, 2005. It was organized by the same members as in the first workshop, with the sponsorship from LINA, the University of Nantes, and the French Association for Constraint Programming, and also with the support by the NII joint research project led by Ueda.

The workshop brought together approximately 30 participants from France and Japan. It was composed of 20 technical talks, ranging over various aspects and applications of constraint programming as in the previous workshop, in addition to an invited talk presented by Mats Carlsson on global constraints.

4 Conclusions and Future Prospects

In this report, we provided our research collaboration between France and Japan in the area of constraint programming. We have been conducting several joint research projects of different kinds. The SCooP project is the general framework for the collaboration between LINA and NII in this area. By contrast, our SAKURA project, funded by Égide and JSPS, treated a more concrete research topic. In addition to the joint research, we

organized two workshops in 2004 and 2005 to enhance collaboration between French and Japanese researchers working on constraint programming.

We are continuing and expanding this Franco-Japanese research collaboration on constraint programming. We are currently setting up a new research project as a successor to the SAKURA project. Although we focused the SAKURA project mainly on constraint solving, we will cover the modeling aspect of constraint programming in the new project. Also, we expect to invite researchers in other areas to the next project, in order to explore new application areas of constraint programming. In addition to the joint research, we wish to continue FJCP workshops in order to help to provide opportunities for further research collaboration between France and Japan.

Acknowledgment

We would like to thank Makoto Yokoo for his continuous cooperation in our activities.

References

- [1] 1st Franco-Japanese workshop on constraint programming (FJCP 2004), October 25–27, 2004, Tokyo, Japan.
<http://research.nii.ac.jp/fjcp/>
- [2] 2nd Franco-Japanese workshop on constraint programming (FJCP 2005), November 14–16, 2005, Le Croisic, France.
<http://fjcp2005.free.fr/>
- [3] F. Benhamou and A. Colmerauer, editors. *Constraint Logic Programming: Selected Research*. MIT Press, 1993.
- [4] M. Ceberio, H. Hosobe, and K. Satoh. Speculative constraint processing with iterative revision for disjunctive answers. In *Computational Logic in Multi-Agent Systems—CLIMA VI*, volume 3900 of *LNAI*, pages 340–357. Springer, 2006.
- [5] A. Colmerauer. An introduction to Prolog III. *Comm. ACM*, 33(7):69–90, 1990.
- [6] M. Gangnet and B. Rosenberg. Constraint programming and graph algorithms. *Ann. Math. Artif. Intell.*, 8(3–4):271–284, 1993.
- [7] R. Horst and P. Pardalos, editors. *Handbook of Global Optimization*. Kluwer Academic Publishers, 1995.
- [8] K. Satoh, P. Codognet, and H. Hosobe. Speculative constraint processing in multi-agent systems. In *Intelligent Agents and Multi-Agent Systems—PRIMA 2003*, volume 2891 of *LNAI*, pages 133–144. Springer, 2003.
- [9] K. Satoh, K. Inoue, K. Iwanuma, and C. Sakama. Speculative computation by abduction under incomplete communication environments. In *Proc. 4th Intl. Conf. on Multi-Agent Systems (ICMAS 2000)*, pages 263–270, 2000.
- [10] M. Wilson. Hierarchical constraint logic programming (Ph.D. dissertation). Technical Report 93-05-01, Dept. Comput. Sci. Eng., Univ. Washington, 1993.