Synergy of Symbolic AI and HPC: Towards a New Generation of Powerful and Reliable AI solving Systems

1 Supervisors

This PhD thesis will be supervised by 3 EC in AI from the LERIA, Angers university, 2 from the RIC theme, and one from the MOC theme:

1. Frédéric Lardeux, RIC, expert in constraint modeling.
2. Eric Monfroy, RIC, expert in collaborative constraint solving and constraint modeling.
3. Jean-Michel Richer, MOC, expert in parallel programming.

2 Background

Symbolic AI covers several sub-domains including Constraint programming (CP). CP is a generic tool for modeling (representing) and solving combinatorial problems. It has found its way into industrial applications. Constraint Satisfaction Problems (CSP) and Constraint Optimization Problems (COP) are often NP-hard and therefore, there is little hope to find a polynomial time algorithm to solve them all. NP-hard problems in general are considered to be intractable. However, that does not prevent people to formulate their (NP-hard) problem as a CSP or a COP, and use a constraint solver to find a good solution within a reasonable amount of time using a clever search heuristic.

High-Performance Computing (HPC) refers to the use of supercomputers or computer clusters to solve complex computational problems that require a large amount of processing power, memory, or storage capacity. One general way of speeding up computations is to exploit the multiple processors of the supercomputers in parallel. By splitting the work evenly over several processors, the result is hoped to be computed in less time, inversely proportional to the number of processors used. Usually, speed-up is less though, because some time is spent on communication. Over the past decades, computer architecture has undergone some major changes and parallelism became essential.

Constraint solvers use a combination of inference and search to solve any given problem. Therefore, parallelization efforts can be classified into one of the above categories:

- Parallel inference, which comprises parallelization of constraint propagation algorithms and parallelization of consistency filtering algorithms,
- Parallel search, which concerns search space splitting, portfolio algorithms, distributed CSPs, and problem decomposition.

Going more onto detail, we can site several approaches based on:

- parallelization architectures, such as GPU's [8], or heterogeneous collections of hardware
- parallel programming model paradigms, such as PGAS (Partitioned Global Address Space), shared memories, or message passing
- type of concurrency, such as concurrent cooperative (such as CCP, Concurrent Constraint programming [7]), or concurrent competitive (such as Adaptive Search [3])

3 Objective

The context of this PhD thesis is thus related to symbolic AI in the framework of parallel constraint programming.

There are several techniques for solving CSPs: constraint programming and propagation-based constraint solvers, linear programming, Boolean satisfaction methods (SAT), metaheuristics, ... We will focus on a method derived from symbolic AI, i.e., constraint propagation-based solvers that are complete solvers (this means that they are able to prove that there is no solution, and in case there is one solution, they are able to find it). The aim of this thesis will be to propose and implement new algorithms for constraint propagation and search for CSP/COP resolution on parallel architectures. The methodology will be to match a formal parallel scheme for propagation (such as distributed chaotic iteration [6, 5] which is an extension of chaotic iteration [2], asynchronous iteration [4], or an extension of concurrent constraint programming [7] for example) to a programming parallel model (such as PGAS [1], shared memory, etc.). Moreover, the idea will be to design a generic solver, not a specific solver for a given problem. The prototype will be validated with tests on standard benchmarks from the literature.

The main phases of the thesis will be:

1. study of existing models
2. design of the formal propagation framework together with the programming parallel model

3. implementation of a prototype
4. validation of the ideas with some benchmarks and comparisons with existing systems

This work will lead to publication in international conferences and journals in AI of good quality. Moreover, several international cooperations are planned.

4 Context

We currently have several international cooperations dedicated to parallelism and constraint programming. This thesis will thus be conducted in this context and will surely lead to some visits to or from these collaborator's institute:

1. With Salvador Abreu (University of Evora and NOVA lab Lisbon, Portugal) we had in 2019 a MIR project, and we currently have a bilateral PHC Pessoa project. We are currently working on a propagation algorithm based on a PGAS model.
2. With Tomasz Jastrab, University of Silesy, Gliwice, Poland, we have a PHC Plonium project. We are currently working on parallel learning of formal language with constraints.
3. We are in contact and collaborate with the Huawei constraint team in Boulogne (A. Lallouet, expert in constraint, and W. Suijlen).
4. With Pierre Talbot of the University of Luxembourg, we collaborate about CCP, asynchronous iterations, and GPU-based solvers. A MIR project has been submitted for visits during the first semester of 2023 (if accepted).
5. With J. Lejeune (LIP6) and C. Prudhomme (LS2N) who are contributing on the ANR project SeMaFoR, we are collaborating on distributed constraint solving, and distributed consensus algorithms.

We will carry on these collaborations with PHC projects, ANR (when possible with foreign countries), and the ultimate goal would be to build a European net with these institutes.

References

[1] George Almasi. *PGAS (Partitioned Global Address Space) Languages*, pages 1539–1545. Springer US, Boston, MA, 2011.

[2] Krzysztof R. Apt. From chaotic iteration to constraint propagation. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *Automata, Languages and Programming, 24th International Colloquium, ICALP'97, Bologna, Italy, 7-11 July 1997, Proceedings*, volume 1256 of *Lecture Notes in Computer Science*, pages 36–55. Springer, 1997.

[3] Yves Caniou, Philippe Codognet, Florian Richoux, Daniel Diaz, and Salvador Abreu. Large-scale parallelism for constraint-based local search: the costas array case study. *Constraints An Int. J.*, 20(1):30–56, 2015.

[4] P. Cousot. Asynchronous iterative methods for solving a fixed point system of monotone equations in a complete lattice. Res. rep. R.R. 88, Laboratoire IMAG, Université scientifique et médicale de Grenoble, Grenoble, France, Sep. 1977. 15 p.

[5] Éric Monfroy. A coordination-based chaotic iteration algorithm for constraint propagation. In Barrett R. Bryant, Janice H. Carroll, Ernesto Damiani, Hisham Haddad, and Dave Oppenheim, editors, *Applied Computing 2000, Proceedings of the 2000 ACM Symposium on Applied Computing, Villa Olmo, Via Cantoni 1, 22100 Como, Italy, March 19-21, 2000. Volume 1*, pages 262–269. ACM, 2000.

[6] Éric Monfroy and Jean-Hugues Réty. Chaotic iteration for distributed constraint propagation. In Barrett R. Bryant, Gary B. Lamont, Hisham Haddad, and Janice H. Carroll, editors, *Proceedings of the 1999 ACM Symposium on Applied Computing, SAC'99, San Antonio, Texas, USA, February 28 - March 2, 1999*, pages 19–24. ACM, 1999.

[7] Vijay A. Saraswat and Martin C. Rinard. Concurrent constraint programming. In Frances E. Allen, editor, *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, San Francisco, California, USA, January 1990*, pages 232–245. ACM Press, 1990.

[8] Pierre Talbot, Frédéric G. Pinel, and Pascal Bouvry. A variant of concurrent constraint programming on GPU. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 3830–3839. AAAI Press, 2022.