RESEARCH STATEMENT

Guohua Liu

1 Introduction

My research is focused on **computational inference** that concerns effective problem solving via inference with data or knowledge. **Constraint programming, mathematical programming, and machine learning** are the main approaches that I have explored for computational inference. In constraint programming, a problem is modeled as a set of constraints and solved by computing the solutions of the constraints. The modeling languages and constraint solvers in constraint programming have shown very promising performance in computational inference. I have been working on the development of a new constraint programming paradigm. Mathematical programming refers to a family of approaches to model and solve optimization problems using algebraic constraints. I have proposed a framework to integrate mathematical programming and constraint programming. Machine learning comprises a variety of approaches to intelligent use of data to improve the performance of computing systems. I am currently investigating the application of constraint and mathematical programming techniques to machine learning.

Meaningful findings from **big data in computational Biochemistry** can significantly improve the state of the art of the area. However, big data introduces exceptionally challenging problems due to its scale, dimensionality, and complexity. I will exploit constraint programming, machine learning, data mining, and database techniques to address these challenges.

In the rest of the statement I will present the plan of my future work, my current work, and my previous research, followed by a summary.

2 Development of Computational Inference for Big Data

1. **Improve machine learning and data mining using constraint programming.** While machine learning and data mining in computational Biochemistry has been intensively studied, the exploration of constraint programming techniques in this area is limited. In fact, many machine learning and data mining problems share the same goal as constraint programming, which is to find an optimal model satisfying a set of constraints. It is only that constraint programming targets any type of constraint satisfaction problems, whereas machine learning and data mining usually focus on specific applications.

   I will exploit a number of constrain programming techniques for machine learning and data mining. (1) As many constraint programming languages have been available, I will explore these languages for problem modeling to facilitate machine learning and data mining. (2) The performance of constraint solvers has been improved significantly in recent years. These solvers will be good candidates to solve the complex combinatorial problems in machine learning and data mining. (3) Hundreds of global constraints have been invented, each of which specifies a frequently used constraint and comes with a dedicated algorithm. They will be used to encode and solve learning or mining problems. (4) optimizers developed in mathematical programming have shown encouraging performance. Their abilities should be further exploited for machine learning and data mining. In fact, the substantial potential of constraint programming techniques for machine learning and data mining has been receiving a lot of attention, as summarized in [2].

2. **Improve constraint programming using machine learning and data mining approaches.** Difficulties arise in applications of constraint programming to real data due to the quality and feature of the data. Machine learning and data mining techniques can alleviate these difficulties by discovering patterns and making predictions.

   I will address a number of difficulties in constraint programming using machine learning and data mining techniques. (1) In real applications, the data set may be ill-defined with errors, noise, or incompleteness, which make it unreliable to problem solving. Machine learning technique will be employed to process ill-defined data and improve the quality of the solutions. (2) The constraints underlying an application may not be known, so it is impossible for users to formulate the constraints explicitly. I will investigate how to obtain these constraints automatically using machine learning and data mining approaches. (3) The potential search space of a set of constraints could be very large, which makes it extremely difficult to find a solution. Machine learning approaches will be studied to enhance constraint solvers so that they can avoid some hopeless search. (4) Most constraint programming systems cannot effectively solve problems concerning probabilities; for example, the probabilistic planning where the objective is to find a plan maximizing the probability of reaching a goal. A number of machine learning approaches will be introduced to constraint programming to help solving these problems. The growing interests in exploiting machine learning and data mining methods for constraint programming have recently been acknowledged in [12].

3. **Exploit database technology for data mining.** Database management systems (DBMS) technology offers many features that make it valuable when implementing data mining applications. I will study using DBMS to: (1) mine datasets that are considerably larger than main memory, since many database themselves are able to handle information, paging, and swapping when necessary; (2) mine complex data types such as image, video, and audio, as emerging object-relational databases can provide such ability; (3) validate discovered patterns using the on-line analytical processing queries; and (4) keep the information used during mining confidential, as data security has been widely implemented in nowadays databases.

4. **Develop an efficient computational inference system.** Developing an efficient computational inference system that integrates the above-discussed techniques is one of my major goal. The system will consist of an expressive programming language, an efficient inference engine, and a complimentary tool kit. The language should be effective for problem modeling; the inference engine will give solutions to the problems; and the tools will be used for a number of tasks such as data prepossessing, program optimization, and debugging.

3 Ongoing Research

I am currently working on **the integration of mathematical programming and constraint programming and their application in machine learning**.

1. **Integration of mixed integer programming and constraint programming.** Numerical optimization problems are ubiquitous in practice but most constraint programming systems cannot solve these problems efficiently when large numerical domains are involved. Meanwhile, many efficient optimizers have been developed for mixed integer programming (MIP). We [9] proposed an integrated paradigm where a problem is solved using a MIP solver, while modeled by a constraint programming language. The integrated paradigm exploits the computational efficiency of the MIP solver and the representational power of the constraint

programming language. In an analogous study, we [3] integrated satisfiability modulo theory (SMT) provers into constraint programming systems, where numerical constraints are computed by the SMT solvers. These results created new opportunities to combine the capacities of different computational inference paradigms.

2. **Bayesian network learning via constraint programming.** Bayesian network (BN) learning is to find a BN that best satisfies a set of data. A BN is a graph that represents the dependencies between variables, for example, diseases and symptoms and a data set consists of the value of the variables. BN can be used to infer the cause of observed facts, for instance, the presence of a disease given the symptoms. While BN has been widely used in machine learning, derive a BN from a data set is quite challenging where the difficulty is to prevent cycles from a BN. We observed that a constraint programming language can compactly encode the acyclicity of BN and MIP solvers can solve the related optimization problems efficiently. We are trying to combine them together for BN learning.

4 Past Research

My doctoral dissertation was focused on **the language and system development for a new constraint programming paradigm**. The paradigm consists of a logic-based constraint programming language and a solver. In this paradigm, a problem is encoded as a program which is a set of constraints, and then solved by computing the solutions of the constraints using the solver. The language is complete, consistent, and, more remarkably, it facilitates nonmonotonic commonsense reasoning where the conclusions made using currently available knowledge may be withdrawn in the presence of additional knowledge. The paradigm has been widely used for data management, decision support, system verification, and so forth [1]. I improved the constraint programming paradigm in the following aspects.

1. **Semantics of the language.** Semantics is a key aspect of the constraint programming language – it defines the intended meaning of a program. However, different semantics were proposed so that a solution to a problem under a semantics may not be a solution to the problem under the others. Insights on the relation of different semantics are critical to problem modeling. My co-author and I [13] formulated a uniform framework for the study of the existing semantics. Then, we [6, 11] applied the framework to a variety of semantics and found that a well-accepted semantics involves circular justification under another more "cautious" semantics. Note that the circular justification may lead to conclusions that are not "sufficiently" supported in reasoning. Finally, we [4, 10] gave a formal account of the difference between these semantics. These results also enabled us to employ an effective constraint satisfaction technique in inference and provided theoretical basis to efficient implementations.

2. **Solver implementation.** A variety of inference engines have been implemented, each of which compute solutions for a certain class of programs. We [11] showed that almost all programs can be transformed into a class of simple programs. Based on the translation, we developed a solver where general programs are transformed into simple ones and solved efficiently. In another work [5, 7], we implemented an adaptive space pruning technique where pruning is dynamically invoked during a search. This approach exploits the power of the pruning while avoiding its unnecessary overhead. In a related work [14], we evaluated the power of the pruning technique via a comparative study with a widely used constraint satisfaction technique.

3. **Program optimization.** Program optimization aims to simplify a program in some aspects. A key property of optimization is that the resulting program must preserve the meaning of the original one. We [8] formulated the criterion under which a program can be optimized while preserving its original meaning in any inference context.

5 Summary

Computational inference techniques are very promising for data analysis in computational Biochemistry. In particular, the combination of constraint programming, machine learning, data mining, and database techniques can provide effective inference systems where the constraint programming techniques provide expressive modeling languages and fast inference engines and the machine learning, data mining, and database approaches help to improve the data quality, problem formulation, and inference efficiency. Intersections of these fields provide well-motivated directions for further investigations and my research agenda sits at their nexus.

References

[1] Gerhard Brewka, Thomas Eiter, and Miroslaw Truszczynski. Answer set programming at a glance. *Commun. ACM*, 54(12):92–103, 2011.

[2] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks.* Cambridge University Press, 2009.

[3] Tomi Janhunen and Guohua Liu Ilkka Niemelä. Strong equivalence of logic programs with abstract constraint atoms. In *GTTV*, pages 161–173, 2011.

[4] G. Liu. Level mapping induced loop formulas for weight constraint and aggregate programs. In *Proc. LPNMR'09*, pages 444–449, 2009.

[5] G. Liu and J. You. On the effectiveness of looking ahead in search for answer sets. In *Proc. LPNMR'07*, pages 303–308, 2007.

[6] G. Liu and J. You. Lparse programs revisited: semantics and representation of aggregates. In *Proc. ICLP'08*, pages 347–361, 2008.

[7] G. Liu and Jia-Huai You. Adaptive lookahead for answer set computation. In *Proc. ICTAI'07*, pages 230–237, 2007.

[8] Guohua Liu, Randy Goebel, Tomi Janhunen, Ilkka Niemelä, and Jia-Huai You. Strong equivalence of logic programs with abstract constraint atoms. In *LPNMR*, pages 161–173, 2011.

[9] Guohua Liu, Tomi Janhunen, and Ilkka Niemelä. Answer set programming via mixed integer programming. In *KR*, 2012.

[10] Guohua Liu and Jia-Huai You. Level mapping induced loop formulas for weight constraint and aggregate logic programs. *Fundam. Inform.*, 101(3):237–255, 2010.

[11] Guohua Liu and Jia-Huai You. Relating weight constraint and aggregate programs: Semantics and representation. *Theory and Practice of Logic Programming*, 2011.

[12] Barry O'Sullivan. Automated modelling and solving in constraint programming. In *AAAI*, 2010.

[13] J. You and G. Liu. Loop formulas for logic programs with arbitrary constraint atoms. In *Proc. AAAI-08*, pages 584–5895, 2008.

[14] J. You, G. Liu, L. Yuan, and C. Onuczko. Lookahead in Smodels compared to local consistencies in CSP. In *Proc. of LPNMR'05*, pages 266–278, 2005.