Constraints: The Ties That Bind

Eugene C. Freuder

Cork Constraint Computation Centre
Computer Science Department
University College Cork
Cork, Ireland
e.freuder@4c.ucc.ie

Abstract

Constraints can serve as a unifying force in artificial intelligence.

Introduction

The Artificial Intelligence community, and the AAAI conference in particular, has long been concerned with counteracting the tendency towards fragmentation, and refocusing on the seminal vision of a unified machine intelligence. The Senior Member Presentation Track itself is one welcome effort in that direction. The purpose of this paper is to delineate the roles that constraint modeling and reasoning can play in pursuing this integrative agenda.

Tutorial

I will provide a very brief introduction to constraint satisfaction using the popular Sudoku game for illustration. Constraints are sometimes viewed primarily in terms of search, but I would argue that their role in representation and reasoning is equally if not more important.

The Sudoku puzzle has recently achieved wide popularity. The puzzle involves a 9 by 9 grid of cells, further divided into nine 3 by 3 blocks of 9 cells each. A given puzzle starts with some cells filled in with one of the digits 1 through 9. The object is to fill the remaining cells with digits such that each row, column and block uses all 9 digits.

This puzzle exhibits the classic features of a *constraint satisfaction problem* (*CSP*). There is a set of problem *variables*, in this case the Sudoku cells. For each variable there is a set of possible *values*, in this case the digits 1 through 9. There are *constraints* that specify which combinations of values are acceptable, in this case the requirements that no two cells in any row, column or block can have the same digit.

A solution to the Sudoku CSP an assignment of digits (values) to each of the empty cells (variables) that satisfies the Sudoku rules (constraints). It is important to recognize that the CSP paradigm has been extended in a variety of directions. It embraces optimisation, uncertainty, change, preferences, continuous sets of values, constraints over many variables, etc.

Of course, CSPs are not restricted to puzzles. Some other examples of constraints:

- The meeting must start at 6:30.
- The separation between the soldermasks and nets should be at least 0.15mm.
- This model only comes in blue and green.
- This cable will not handle that much traffic.
- John prefers not to work on weekends.
- The demand will probably be for more than 5 thousand units in August.

Some examples of constraint satisfaction or optimization problems:

- Schedule these employees to cover all the shifts.
- Optimize the productivity of this manufacturing process.
- Configure this product to meet my needs.
- Find any violations of these design criteria.
- Optimize the use of this satellite camera.
- Align these amino acid sequences.

If you have solved a Sudoku puzzle you have probably used some of the basic constraint-based reasoning and search techniques. For example, saying that "there can't be a 7 in this cell, because there already is a 7 in that cell" is an example of "arc consistency constraint propagation".

A couple of basic references are Dechter's textbook on *Constraint Processing*, and the new *Handbook of Constraint Programming*, edited by Rossi, van Beek, and Walsh.

History

Constraints have played a role in AI from the early days when pioneering programs first exhibited facets of human intelligence. Much of the earliest work on constraints in AI was conducted in the context of systems and applications:

- Constraints played a role in Ivan Sutherland's seminal 1963 MIT Ph.D. thesis, "Sketchpad: A man-machine graphical communication system".

- In the very first issue of *Artificial Intelligence* in 1970, constraint satisfaction and propagation is included in Fikes' "REF-ARF: A System for Solving Problems Stated as Procedures".

- Kowalski employed a form of constraint propagation for theorem proving.

- Sussman and others at MIT applied a form of constraint propagation to circuit analysis, synthesis, and fault localization.

- Constraints were used by Sussman with Steele in their CONSTRAINTS language, by Borning in his ThingLab simulation laboratory, whose kernel was an extension of the Smalltalk language, and by Lauriere in his language for solving combinatorial problems, Alice.

- Eastman did "constraint structured" space planning with GSP, the General Space Planner, Stefik used "constraint posting" in MOLGEN to plan gene-cloning experiments in molecular genetics, and Descotte and Latombe employed a planner which made compromises among "antagonistic constraints" in their GARI system to generate the machining plans of mechanical parts. Fox, Allen and Strohm used constraint-directed reasoning in ISIS-II for factory job-shop scheduling.

- A great deal of work on constraints developed out of the machine vision community, with early contributors including Montanari, Waltz, Huffman, Clowes, Mackworth, Freuder, Barrow, Tenenbaum, Rosenfeld, Hummel, Zucker, Harlick, Davis, Milgram, Shapiro, and Ullman.

Integration

Constraints can continue to play an integrating role in AI in a number of ways.

Constraints can be used to facilitate other aspects of AI:

- Many forms of reasoning have been modeled as constraint satisfaction:
  - temporal
  - spatial
  - analogical
  - diagnostic
  - qualitative
  - deductive
  - defeasible
  - causal
  - uncertain

- Components of different 'reasoning architectures' have been modeled as constraint satisfaction:
  - case-based
  - rule-based
  - agent-based

- Various facets of human behavior have been modeled as constraint satisfaction:
  - vision
  - language
  - planning

On the other hand, various forms of AI can be brought to bear on constraints:

- A number of "intelligence architectures" have been used for constraint satisfaction:
  - neural networks
  - genetic algorithms
  - evolutionary algorithms

- Advances on several facets of AI have been used for constraint satisfaction:
  - learning
  - truth maintenance
  - reasoning under uncertainty
  - reasoning about preferences
  - case-based reasoning
  - deduction
  - game playing

A grid of the applications of constraints to other areas of AI and vice versa would be heavily filled in, but would reveal opportunities for further benefits, e.g. constraints have been heavily used for planning, but planning has not been heavily used for constraint solving. More needs to be done in bringing to bear AI work on knowledge capture, explanation, intelligent user interfaces, expert systems, and automated software engineering to promote ease of use for constraint programming.

Constraints have been used to solve a wide variety of problems in a great many fields:

- telecommunications
- electronic commerce
- electronic, mechanical, civil engineering
- scheduling and planning
- testing and verification
- bioinformatics
- transportation
- electronic games
- music
- retail
- manufacturing
- natural language processing
- speech recognition
- computer vision
- design and configuration
- grid computing
- forestry
- finance
- logistics
- medicine
- construction
- graphics
- farming
- timetabling and rostering
- autonomous vehicles
- storage management
- interoperability
- feature interaction
- food industry
- compilers
- resource allocation
- robotics
- security
- control systems
- forest fires
- utilities
- decision support
- network management
- supply chain management

A good introduction to the applications of constraints is Wallace's paper on "Practical Applications of Constraint Programming" in the first issue of the *Constraints* journal.

Constraints bring in connections to fields beyond AI as well:

- Operations Research
- Combinatorial Algorithms
- Constraint Programming Languages
- Constraint Databases
- Cognitive Science

We can move between or combine constraint-based methods and other approaches from AI or other fields. Constraint-based models can be compiled into other models or vice versa:

- rules
- SAT
- mathematical programming
- automata
- decision trees
- metaheuristics
- binary decision diagrams

There is much more that might be done with hybrid or multi-modal reasoning systems that combine different forms of reasoning in a manner that best suits the application. There is considerable interest currently, for example, in combining methods from constraint programming and operations research.

Examples

I will cite a few examples from our recent work at the Cork Constraint Computation Centre (often in collaboration with others) that illustrate some of the forms of integration and application outlined above. Full citations are available at www.4c.ucc.ie.

- Machine learning and SAT: Acquiring Constraint Networks Using a SAT-based Version Space Algorithm, by Bessiere, Coletta, Koriche, and O'Sullivan in the AAAI-06 Nectar Track.

- Case-based reasoning: Using CBR to Select Solution Strategies to Constraint Programming, Gebruers, Hnich, Bridge, and Freuder at the 2005 International Conference on Case-based Reasoning.

- Game-tree search: Adversarial Constraint Satisfaction by Game-tree Search, by Brown, Little, Creed, and Freuder at the 2004 European Conference on Artificial Intelligence.

- Cognitive architectures: Learning to Support Constraint Programmers, by Epstein, Freuder, and Wallace in *Computational Intelligence*.

- Machine learning: Applying Machine Learning to Low Knowledge Control of Optimization Algorithms, by Carchrae and Beck in *Computational Intelligence*.

- Agents: Constraint-based Reasoning and Privacy/efficiency Tradeoffs in Multi-agent

Problem Solving, by Wallace and Freuder in *Artificial Intelligence*.

- Knowledge Capture: Knowledge Base Reuse Through Constraint Relaxation, by Nordlander, Brown, and Sleeman at the 2005 International Conference on Knowledge Capture.

- Uncertainty: Uncertain Linear Constraints, by Wilson at the 2004 European Conference on Artificial Intelligence.

- Preferences: Consistency and Constrained Optimisation for Conditional Preferences, by Wilson at the 2004 European Conference on Artificial Intelligence.

- Security: A Soft Constraint-based Approach to the Cascade Vulnerability Problem, by Bistarelli, Foley, and O'Sullivan in the *Journal of Computer Security*.

- Testing: Constraint Models for the Covering Test Problem, by Hnich, Prestwich, Selensky, and Smith in the *Constraints* journal.

- Scheduling: Supporting Dispatchability in Schedules with Consumable Resources, by Wallace and Freuder in the *Journal of Scheduling*.

- Networks: Symmetry and Search in a Network Design Problem, by Smith at the 2005 International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems.

- Inventory: Computing (R_t,S_t) Policies in a Non-Stationary Stochastic Demand Inventory System Using Constraint Programming, by Tarim and Smith at the 2005 Conference of the European Chapter on Combinatorial Optimization.

- Auctions: Robust Solutions for Combinatorial Auctions, by Holland and O'Sullivan at the 2005 ACM Conference on Electronic Commerce.

- Design: Supporting Constraint-Aided Conceptual Design from First Principles in Autodesk Inventor, by Holland, O'Callaghan, and O'Sullivan at the 2005 International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems.

Workshops

I present here some pointers to workshops that have focused on Constraints and X, for a variety of 'X':

- Planning
http://www.aaai.org/Press/Reports/Workshops/ws-00-02.php

- Verification
http://www.newton.cam.ac.uk/programmes/LAA/laaw05.html

- Language Processing
http://www.lpl.univ-aix.fr/~blache/cslp06/index.html

- Design
http://www.cs.ucc.ie/~osullb/cad05/

- Bioinformatics
http://www.dimi.uniud.it/dovier/WCB05/

- Music
http://recherche.ircam.fr/equipes/repmus/cpws/

- Control
http://www2.parc.com/spl/projects/constraints-in-control/cc99/

- Security
http://www.sci.unich.it/~bista/organizing/CPSec/

- Agents
http://www.aaai.org/Library/Workshops/ws97-05.php

Conclusion

The fact that so many forms of reasoning and so many applications can be modeled as constraint satisfaction suggests that constraints can have a role to play as a *lingua franca* for expressing and integrating approaches to human-level intelligence, and in providing a common ground for their study. A "deep structure" constraint model may be used to support alternative forms of reasoning and alternative applications.

About ten years ago I wrote a paper "Constraint Programming: In Pursuit of the Holy Grail" in which I suggested constraint programming as an appropriate vehicle for the ultimate pursuit in computer science: a computer that simply accepts your statement of your problem and proceeds to solve it. Of course, that goal is not to be reached in ten years. However, an integrated application of AI methods can provide the fuel that enables the vehicle of constraint programming to drive us further towards that goal.

Acknowledgments

This material is based upon works supported by the Science Foundation Ireland under Grant No. 00/PI.1/C075.