CSCE 421/821: Foundations of Constraint Processing, Spring 2009

List of Projects

Berthe Y. Choueiry
Constraint Systems Laboratory
Department of Computer Science and Engineering
University of Nebraska-Lincoln
choueiry@cse.unl.edu
February 19, 2009

Distributed: Friday, February 20, 2009

Project selection: Friday, February 27, 2009 (by handin)

Progress reports: Friday, March 27, 2009 (by handin)

Final reports: Friday, April 24, 2009 (paper and by handin)

Presentations: M/W/F, April 27/29/ May 1, 2009 (evening sessions scheduled as necessary)

Code and slides: Friday, May 1, 2009 (by handin)

Notes: *You must clearly acknowledge help received from anyone*. Always acknowledge sources of information (URLs, books, classnotes, etc.).

Contents

---

1 Guidelines

This is a non-exhaustive list of possible topics for semester projects. *If you have an idea for a project, do not hesitate to discuss it with the instructor.* There are three main categories to choose from:

1. A project from one of the following alternatives. These include the following main alternatives:
   - Implement and evaluate an algorithm, Sections 2.
   - Study modeling and problem formulation, Section 3.
   - Investigate an advanced theoretical concept, Section 4.
2. Conduct a critical literature survey, Section 5.
3. The search challenge described in Section 6. This challenge requires familiarity with Common Lisp or Java.

Finally, a few suggestions for paper summaries and presentations to improve your grade are provided in Section 7.

More guidelines:

- Some projects may have enough substance to be conducted in a team of two students. When this is the case, *each student will have* to provide the instructor with an evaluation of the performance of his/her team partners. This feedback could be provided orally or by filling a standard form (ask instructor for the form).
- The same project may be chosen by more than one person or team. So, do not rush to 'reserve' yourself a project. If a project is selected by more than one person or team, we will carry out a comparison of the approach and results.
- Again, you are encouraged to design your own project proposal and discuss it with the instructor.
- Most CP papers seem to be now available online from Springer, which did not use to be the case until recently. It *may* be the case that you have to be on campus.

Projects markes with a * are new additions from the list of previous years.

---

2 Experimental evaluation of advanced algorithms

Every implementation should be tested on a real-world problem (when available), randomly generated problems, or both. Results should be reported in terms of nodes visited, constraint checks, CPU time, and other applicable criteria. Generators for random CSPs are available. Tests should be conducted for various values of constraint tightness and density, and results should be averaged for at least 50 problem instances per measurement point. Details of the testing and evaluation methodologies should be discussed with the instructor on a case-by-case basis.

1. **Ordering heuristic.* Study, implement, and evaluate the ordering heuristics techniques proposed by Refalo [Refalo 2004].
2. *Ordering heuristic.* Study, implement, and evaluate the ordering heuristics techniques proposed by Zanarini and Pesant [2007] and their application to `alldifferent` and `regular` constraints.
3. *Distributed CSPs.* Study, implement, and evaluate an algorithm for asynchronous backtracking such as the one by Zivan et al. [2007], Zivan and Meisels [2005], or by Maestre and Bessière [2004].
4. **Propagation algorithms: arc consistency* Study, implement, and evaluate the family of propagation algorithms AC-* proposed by Régin [2005].

5. **Propagation algorithms: path consistency.* Implement and compare the performance of the following algorithms for path consistency: `PC-2` [Mackworth 1977], `PC-8` [Chmeiss and Jégou 1998], anc `PC-2001` [Bessière *et al.* 2005]. Note: The student who takes this project can collaborate with Peter Schlette who is working on the following project, likely reusing the same data structures as Peter.

6. *Propagation algorithms: path consistency.* Implement and compare the performance of the following algorithms for path consistency: `PC-2` [Mackworth 1977], `DPC` [Dechter 2003], `PPC` [Bliek and Sam-Haroud 1999], and $\triangle$-`PPC` [Xu 2003].

7. *Propagation algorithms: path consistency.* Study, implement, and evaluate the algorithms for Path Consistency by Dual Consistency proposed by Lecoutre et al. [2007a].

8. **Propagation algorithms for the Latin Square.* Implement and compare the performance and pruning power of the following algorithms for solving the Quasigroup Completion Problem: AC, SAC, GAC, and SGAC. The student would be able to use and improve our current implementation for the Sudoku puzzle in Java.

9. **Propagation algorithms for the Kakuro.* Implement and compare the performance and pruning power of the propagation algorithms for solving the Kakuro puzzle [Simonis 2008; Cambazard]. The student may want to use and improve our current implementation for the Sudoku puzzle in Java.

10. **Propagation algorithms for the Latin Square.* Implement and compare the performance and pruning power of the following algorithms for solving the Quasigroup Completion Problem: AC, SAC, GAC, and SGAC. The student would be able to use and improve our current implementation for the Sudoku puzzle in Java.

11. **Parametrized lookahead.* Implement and compare the performance of FC, MAC, and p-MAC for solving binary CSPs. p-MAC is a lookahead algorithm proposed by a student of CSCE821 during Spring 2008 for controlling the depth of the lookahead accorind to a depth parameter p. The student will have to implement AC2001 as a basis for the lookahead. Please discuss with instructor if interested.

12. *Propagation algorithms: global constraints*. Study, implement, and evaluate the propagation algorithm for the 'deviation' global constraint proposed by Schaus et al. [2007].

13. *Propagation algorithms: Temporal Reasoning with qualitative constraints.* Study, implement, and test the algorithms for computing the minimal network on point algebra constraints proposed by Gerevini and Saetti [2007].

14. *Propagation algorithms: subgraph isomorphism.* Study, implement, and test the algorithms for filtering subgraph isomorphism proposed by Zampelli et al. [2007]. (Optional: Compare with a technique for the same purpose studied by the instructor.)

15. *Temporal Reasoning with metric constraints.* Study, implement, and test the algorithms for solving the DTP proposed by Kumar [2005].

16. *Temporal Reasoning with metric constraints: Search.* Study, implement, and evaluate the backtrack search for solving the Disjunctive Temporal Problem (DTP) proposed in Tsamardinos and Pollack [2003].

17. *Symmetry.* Study, implement, and test the techniques proposed by Meseguer and Torras [2001] for exploiting symmetries in backtrack search.

18. *Symmetry.* Study, implement, and test the techniques proposed by Law et al. [2007] for breaking symmetries of interchangeable variables and values.

19. *Dominance.* Study, implement, and test the techniques proposed by Lecoutre et al. [2007b] for improving backtrack search by detecting dominance and pruning the search tree.

20. *Bundling strategies.* (Initial code exists in Common Lisp.) Evaluate bundling strategies on benchmark problems. Modify the MAC algorithm to exploit interchangeability. Implement your modification, test it on binary and non-binary CSPs and compare results with the non-modified algorithm (i.e., MAC without bundling) on benchmark problems.

21. *Backjumping on QCSPs.* Study, implement, and evaluate the algorithm backjumping for Quantified CSPs proposed by Bacchus and Stergiou [2007].

22. *Decomposition.* Study, implement, and evaluate the algorithm for decomposing CSPs of [Chmeiss *et al.* 2003].

3 Modeling

1. *Modeling Software Engineering tasks as CSPs.* Either study the modeling of Program Verification as a CSP proposed by Collaviza and Rueher [2007], or propose a model of this or another task in Software Engineering as a CSP. Implement, test, and evaluate the proposed techniques on toy and/or benchmark problems. Alert: Instructor is not familiar with work in this area.

2. *Global constraints.* Research a number of global constraints, study their semantics, and investigate, implement, and evaluate the specialized propagation algorithms proposed in the literature for this purpose. You may refer to the catalogue in `http://www.emn.fr/x-info/sdemasse/gccat/` or proceedings of main conferences.

4 Research

1. *Consistency*. Prove that SGAC can solve any well-posed 9×9 well-posed Sudoku CSPs.

2. *Theoretical aspects of Constraint Satisfaction.* Study the work of Atserias et al. [2007] and Feder and Vardi.

3. *Use of Database techniques in CSPs.* Study how the join ordering strategies (e.g., using dynamic programming) can be exploited for finding all the solutions to a non-binary constraint satisfaction problem. Implement, test, and evaluate the selected algorithm.

4. *Generation of random solutions.* Study, implement and test the method for generating solutions to a CSP uniformly at random of [Dechter *et al.* 2002].

5 Literature review

Conduct a critical review and a synthesis of an area where constraints are studied or applied, such as:

1. Symmetric CSPs.

2. Hypertree decomposition (tractability studies of CSPs), e.g. [Hubie Chen 2005; Grohe and Marx 2006].

3. Use of SAT and constraints in Model Checking.

4. Soft constraints and preferences in CSPs: CP-nets.

5. Temporal reasoning.

6. Distributed CSPs.

7. Numeric (a.k.a. continuous) CSPs.

8. Relationship between CSPs and belief networks (recent work by Dechter).

6 Search competition

In the Constraint Systems Laboratory, we have modeled as a CSP the problem of assigning Graduate Teaching Assistants (GTAs) to courses in the department [Glaubius 2001; Glaubius and Choueiry 2002; Zou and Choueiry 2003; Lim *et al.* 2004; Guddeti and Choueiry 2004]. We have encoded the model (i.e., data structures and constraints) in Common Lisp, but a (partial) implementation in Java exists. Students are requested to design and implement their own problem-solving strategy to handle this over-constrained optimization problem. Results will be compared according to a predefined set of optimization criteria. We will provide a specification of the problem, a collection of data, and a (sketchy) manual of the main functions for checking consistency of constraints. Students can investigate any of the following approaches:

1. Learning techniques for the development of an algorithm-selection mechanism: we have (at least) 5 search techniques implemented. The goal is to build a mechanism of portofolio selection based on the performance of these algorithms, building on the work of [Leyton-Brown *et al.* 2003b; 2003a; for Automatic Algorithm Portfolio Seclection 2004].

2. Constructive search.

3. Iterative repair (e.g., hill-climbing and tabu search).

4. Auction and market-based algorithms.

5. A constructive or local search technique that exploits symmetries.

This project requires the ability to deal with (i.e., program in or interface with) Common Lisp. It also requires a sense of creativity for designing a new problem-solving strategy, or choosing and adapting one from the literature. Since the current prototype is the product of research activities, the robustness and stability of the code are not guaranteed.

7 Some papers for summaries or presentations

Here is a list of papers you may want to study for a summary:

- *Various papers on industrial applications of CP in telecommunications, planning and scheduling in a large oil pipeline network, protein structure prediction, etc. from the Proceedings of CP 2008.

- *Tractability of perfect constraints [Salamon and Jeavons 2008].

- Semiring and soft constraints for diagnosis [Sachenbacher and Williams 2004].
- The paper in conditional interchangeability and substitutability of [Zhang and Freuder 2004].
- The paper on variable ordering heuristics by Beck et al. [2003]. (You may find it useful to read the short ones too: [Beck *et al.* 2004a; 2004b].

References

Atserias, Albert; Bulatov, Andrei A.; and Dalmau, Víctor 2007. On the Power of k-Consistency. In *34th International Colloquium on Automata, Languages and Programming (ICALP)*, volume LNCS 4596. Springer. 279–290.

Bacchus, Fahiem and Stergiou, Kostas 2007. Solution Directed Backjumping for QCSP. In 13^{th} *International Conference on Principle and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 148–163.

Beck, J. Christopher; Prosser, Patrick; and Wallace, Richard J. 2003. Toward Understanding Variable Ordering Heuristics for Constraint Satisfaction Problems. In *Proceedings of the Fourteenth Irish Artificial Intelligence and Cognitive Science Conference (AICS03)*.

Beck, J. Christopher; Prosser, Patrick; and Wallace, Richard J. 2004a. Failing First: An Update. In *Proceedings of the Sixteeth European Conference on Artificial Intelligence*. 959–960.

Beck, J. Christopher; Prosser, Patrick; and Wallace, Richard J. 2004b. Variable Ordering Heuristics Show Promise. In *the Tenth International Conference on Principles and Practice of Constraint Programming (CP'04)*.

Bessière, Christian; Régin, Jean-Charles; Yap, Roland H.C.; and Zhang, Yuanlin 2005. An Optimal Coarsegrained Arc Consistency Algorithm. *Artificial Intelligence* 165:165–185.

Bliek, Christian and Sam-Haroud, Djamilla 1999. Path Consistency for Triangulated Constraint Graphs. In *Proc. of the 16* th *IJCAI*, Stockholm, Sweden. 456–461.

Cambazard, Hadrien ard. Basic Understanding of The Propagation Performed in a Constraint Solver Through the Kakuro Puzzle: Insights About Bound Consistency And Generalized Arc Consistency. http://4c.ucc.ie/ hcambaza/page1/page7/page7.html.

Chmeiss, Assef and Jégou, Philippe 1998. Efficient Path-Consistency Propagation. *International Journal on Artificial Intelligence Tools* 7(2):121–142.

Chmeiss, Assef; Jégou, Philippe; and Keddar, Lamia 2003. On a Generalization of Triangulated Graphs for Domains Decomposition of CSPs. In *Proc. of the 18* th *IJCAI*, Acapulco, Mexico. 203–208.

Collavizza, Hélène and Rueher, Michel 2007. Exploring Different Constraint-Based Modelings for Program Verification. In 13^{th} *International Conference on Principle and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 49–63.

Dechter, Rina; Kask, Kalev; Bin, Eyal; and Emek, Roy 2002. Generating Random Solutions for Constraint Satisfaction Problems. In *Proc. of AAAI-2002*, Edmoton, Alberta, Canada. 15–21.

Dechter, Rina 2003. Constraint Processing. Manuscript, forthcoming.

Automatic Algorithm Portfolio Seclection, Learning Techniquesfor 2004. Failing First: An Update. In *Proceedings of the Sixteeth European Conference on Artificial Intelligence*. 475–479.

Gerevini, Alfonso and Saetti, Alessandro 2007. Efficient Computation of Minimal Point Algebra Constraints by Metagraph Closure. In 13^{th} *International Conference on Principle and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 301–316.

Glaubius, Robert and Choueiry, Berthe Y. 2002. Constraint Modeling and Reformulation in the Context of Academic Task Assignment. In *Working Notes of the Workshop on Modelling and Solving Problems with Constraints, ECAI 2002*, Lyon, France.

Glaubius, Robert 2001. A Constraint Processing Approach to Assigning Graduate Teaching Assistants to Courses. Undergraduate Honors Thesis. Department of Computer Science and Engineering, University of Nebraska-Lincoln.

Grohe, Martin and Marx, Dániel 2006. Constraint Solving via Fractional Edge Covers. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. ACM. 289–298.

Guddeti, Venkata Praveen and Choueiry, Berthe Y. 2004. An Empirical Study of a New Restart Strategy for Randomized Backtrack Search. In *Workshop on Immediate Applications of CSPs (CP 04)*, Toronto, Canada. 66–82.

Hubie Chen, Víctor Dalmau 2005. Beyond Hypertree Width: Decomposition Methods Without Decompositions. In *11th International Conference on Principles and Practice of Constraint Programming (CP 05)*, volume LNCS 3709. Springer. 167–181.

Kumar, T. K. Satish 2005. On the Tractability of Restricted Disjunctive Temporal Problems. In *ICAPS 2005*. AAAI Press. 110–119.

Law, Y.C.; Lee, J.H.M.; Walsh, Toby; and Yip, J.Y.K. 2007. Breaking Symmetry of Interchangeable Variables and Values. In 13^{th} *International Conference on Principle and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 423–437.

Lecoutre, Christophe; Cardon, Stéphane; and Vion, Julien 2007a. Path Consistency by Dual Consistency. In 13^{th} *International Conference on Principle and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 448–452.

Lecoutre, Christophe; Sais, Lakhdar; Tabary, Sébastien; and Vidal, Vincent 2007b. Exploiting Past and Future: Pruning by Inconsistent Partial State Dominance. In 13^{th} *International Conference on Principle and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 453–467.

Leyton-Brown, K.; Nudelman, E.; Andrew, G.; McFadden, J.; and Shoham, Y. 2003a. Boosting as a metaphor for algorithm design (working paper). http://www.cs.ubc.ca/˜kevinlb/papers/boosting.ps.

Leyton-Brown, K.; Nudelman, E.; Andrew, G.; McFadden, J.; and Shoham, Y. 2003b. Boosting as ametaphor for algorithm design. In *Constraint Programming*.

Lim, Ryan; Guddeti, Venkata Praveen; and Choueiry, Berthe Y. 2004. An Interactive System for Hiring and Managing Graduate Teaching Assistants. In *Conference on Prestigious Applications of Intelligent Systems (ECAI 04)*, Valencia, Spain. 730–734.

Mackworth, Alan K. 1977. Consistency in Networks of Relations. *Artificial Intelligence* 8:99–118.

Maestre, Arnold and Bessière, Christian 2004. Improving Asynchronous Backtracking for Dealing with Complex Local Problems. In *European Conference on Artificial Intellifence (ECAI 04)*, Valencia, Spain. 206–210.

Meseguer, Pedro and Torras, Carme 2001. Exploiting Symmetries Within Constraint Satisfaction Search. *Artif. Intell.* 129((1-2)):133–163.

Refalo, Philippe 2004. Impact-Based Search Strategies for Constraint Programming. In *9th International Conference on Principles and Practice of Constraint Programming (CP 04)*, volume LNCS 3258. Springer. 557–571.

Régin, Jean-Charles 2005. AC-*: A Configurable, Generic and Adaptive Arc Consistency Algorithm. In *11th International Conference on Principles and Practice of Constraint Programming (CP 05)*, volume LNCS 3709. Springer. 505–519.

Sachenbacher, Martin and Williams, Brian C. 2004. Diagnosis as Semiring-Based Constraint Optimization. In *European Conference on Artificial Intellifence (ECAI 04)*, Valencia, Spain. 873–877.

Salamon, András Z. and Jeavons, Peter G. 2008. Perfect Constraints Are Tractable. In 14^{th} *International Conference on Principle and Practice of Constraint Programming (CP 08)*, volume 5202 of *Lecture Notes in Computer Science*. Springer. 524–528.

Schaus, Pierre; Deville, Yves; and Dupont, Pierre 2007. Bound-Consistent Deviation Constraint. In 13^{th} *International Conference on Principle and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 620–634.

Simonis, Helmut 2008. Kakuro as a Constraint Problem. In *the* 7^7*th International Workshop on Constraint Modelling and Reformulation*. 1–15.

Tsamardinos, Ioannis and Pollack, Martha E. 2003. Efficient Solution Techniques for Disjunctive Temporal Reasoning Problems. *Artificial Intelligence*. In press.

Xu, Lin 2003. Consistency Methods for Temporal Reasoning. Master's thesis, Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE. May 2003.

Zampelli, Stéphane; Deville, Yves; Solmon, Christine; Sorlin, Sébastien; and Dupont, Pierre 2007. Filtering for Subgraph Isomorphims. In 13^{th} *International Conference on Principle and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 728–742.

Zanarini, Alessandro and Pesant, Gilles 2007. Solution Counting Algorithms for Constraint-Centered Search Heuristics. In *13th International Conference on Principles and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 743–757.

Zhang, Yuanlin and Freuder, Eugene C. 2004. Conditional interchangeability and substitutability. In *Fourth International Workshop on Symmetry and Constraint Satisfaction Problems (SymCon'04)*, Toronto, Canada.

Zivan, Roie and Meisels, Amnon 2005. Dynamic Ordering for Asynchronous Backtracking on DisCSPs. In *11th International Conference on Principles and Practice of Constraint Programming (CP 05)*, volume LNCS 3709. Springer. 32–46.

Zivan, Roie; Zazone, Moshe; and Meisels, Amnon 2007. Min-Domain Ordering for Asynchronous Backtracking. In *13th International Conference on Principles and Practice of Constraint Programming (CP 07)*, volume 4741 of *Lecture Notes in Computer Science*. Springer Verlag. 758–772.

Zou, Hui and Choueiry, Berthe Y. 2003. Characterizing the Behavior of a Multi-Agent Search by Using it to Solve a Tight, Real-World Resource Allocation Problem. In *Workshop on Applications of Constraint Programming*, Kinsale, County Cork, Ireland. 81–101.