Chalmers University of Technology
Department of Computing Science

Artificial Intelligence
lp 4 VT12

Short Introduction to Constraint Programming

March 30, 2012

Birgit Grohe

What is this lecture about?

- The Constraint Satisfaction Problem (CSP)
- Examples
- Solution methods
  - Consistency techniques
  - Backtracking search
- Constraint optimization problems (COP)

The Constraint Satisfaction Problem (CSP)

- A set X of **variables** $X = \{X_1, \ldots, X_n\}$,
- a set D of discrete, finite **domains** for each variable $D(X_i) = \{v_1, \ldots, v_d\}$,
- and a set of **constraints** C over subsets of variables.

A **solution** to a CSP is an assignment of values to each variable such that no constraint is violated.

Constraints

- $X_1 + 2X_2 \geq 4$ (linear constraint)
- $5X_1 \cdot X_2 + X_1^3 = 7$ (nonlinear constraint)
- $X_1 \vee X_2 \vee X_3$ (logical constraint)
- $alldiff(X_1 \ldots X_n)$ (so called *global constraints*)
- general constraints (table)
- ...

Example CSP: Cryptarithmetic Puzzle

Each letter stands for a digit, find a substitution of digits to letters such that the resulting sum is arithmetically correct. TWO + TWO = FOUR

Variables: $X = \{T, W, O, F, U, R\} \cup \{C_{10}, C_{100}, C_{1000}\}$

Domains: $D(X_i) = \{0, 1, \ldots, 9\}$, $D(C_i) = \{0, 1\}$

Constraints:

$$\begin{aligned} O + O &= R + 10C_{10} \\ W + W + C_{10} &= U + 10C_{100} \\ T + T + C_{100} &= O + 10C_{1000} \\ C_{1000} &= F \\ F, T &\geq 1 \end{aligned}$$

$alldiff(T, W, O, F, U, R)$

Russel-Norvig, p207, figure 6.2

Example CSP: Map Coloring

Given a map of Australia, color each region in red, green or blue such that neighboring regions always have different colors.

Russel-Norvig, p204-205, figure 6.1

Variables: $X = \{WA, NT, Q, NSW, V, SA, T\}$

Domains: $D = \{red, green, blue\}$ for each variable

Constraints: $C = \{SA \neq WA, SA \neq NT, SA \neq Z, SA \neq V, SA \neq VSW, WA \neq NT, \ldots\}$

CSPs with binary variables can easily be visualized in a **constraint graph** where the variables are nodes and the constraints are edges.

How to Solve a CSP?

CSP's are usually (NP-)hard problems.

- **Search**
  - Backtracking (DFS-style search).
- **Constraint propagation**
  Operates locally on one constraint and its involved variables:
  - Consistency techniques.

Other Examples for CSPs

- Map coloring problem, *(famous 4-color theorem)*
- Cryptarithmetic puzzles, N-queens problem, Sudoku,
- Satisfiability Problem (SAT) *(link to logic and planning!)*
  - Electronic design automation: model checking, formal verification of microprocessors, routing of FPGAs, etc.
- Day-of-operation problem in the airline industry, *Operations Research (OR)*
- Job-shop scheduling,
- ...

Consistency Techniques

Usage of consistency techniques is sometimes called *inference* or *(constraint) propagation*.

Node consistency: Given a constraint with one variable, remove values from the domain that are inconsistent with the constraint.

Example cryptarithmetic puzzle: $F \geq 1$ removes the value 0 from the domain of F.

Arc consistency: Given a constraint with two variables, remove values inconsistent with the constraints from both variable-domains.

Example: $C_{1000} = F$ removes value 0 from the domain of C_{1000} and values $2, \ldots, 9$ from the domain of F. (Thus we get $F = 1$.)

Consistency Techniques

It can be proved that each constraint with more than 2 variables can be expressed by a set of binary constraints and auxiliary variables.

Therefore, the CP-community has until quite recently focused on **consistency techniques for binary constraints:** Algorithms AC-1, AC-3 etc.

Also methods for path consistency and k-consistency build on binary constraints, but considering $2 \ldots k$ constraints at a time.

Fashionable ***global constraints*** like $alldiff$ have revolutionized consistency and modeling techniques: speed up for the propagation, decreased need of search, larger problems can be solved.

Design Issues for Backtracking Search

- **Variable ordering**
  - *first-fail* (*most constrained variable first / minimum remaining value first*)
  - *largest degree first* (constraint graph)
- **Value ordering**
  - *least-constraining value*

Good variable and value orderings are very often based on experience. (→ evaluation order in alpha-beta search).

What about Optimization instead of Satisfiability?

Map coloring:
Instead of *find a coloring with a given number of colors* solve *find a coloring with the minimum number of colors*.

For a linear objective function, linear constraints and continuous and/or integer variables there are good methods already.

- $\min\{cx | Ax = b, x \geq 0\}$ Linear Programing (LP) - The Simplex method
- $\min\{cx | Ax = b, x \text{ integer }\}$ Integer Linear Programming (ILP) - Branch & Bound, Cutting Planes etc.

Constraint Optimization Problems (COP)

If we want to use non-linear and global (e.g. alldiff) constraints, then an objective function can be modeled as a constraint of the type

$$constr \leq MAXCOST$$

and if the CSP is satisfiable, decrease MAXCOST, otherwise increase.

Can be quite inefficient compared to LP and ILP methods!

Cost Propagation

If we allow *preference constraints*, then consistency techniques are not applicable.

Numerical propagation can be used to spread the preference-values in the constraint graph to find the optimal solution.

Example: Given a crossword grid filled in with coded letters, find an assignment from coded to uncoded letters so that all words are valid English words.

Crossword Puzzle

Variables: $X_{ij} \in \{0,1\}$ is 1 if coded letter i corresponds to uncoded letter j, 0 otherwise.

Constraints: $\sum X_{ij} = 1 \forall i, j$ one-to-one mapping of letters.

Preference constraints: Pairwise constraints for neighboring letters in the grid according to bigram-statistics.

Objective function/unary constraints: For each X_{ij} according to letter frequency.

Add. material: Letter and bigram frequency, crossword puzzle

Crossword Puzzle

It turns out that this Crossword puzzle can be solved with propagation only. No search is needed.

Furthermore, the preference constraints look very much like the tables in problabilistic inference and the calculations are very similar as well!

On a more general level one could say that this kind of inference is about exchanging information between entities until (hopefully) a consensus about the best solution is found.

Constraint Programming - Summary

- CSP-solvers are *general purpose methods*, independent of problem specific features.
- Constraint programming allows to express a large range of toy and real world problems.
- Easy problems can be solved with propagation only.
- Difficult problems need a carefully chosen mix of search and propagation.
- Suitable **modeling** is crucial. Many global and nonlinear constraints tend to slow down the solution process.

Commercial CSP solver: ILOG (http://www.ilog.com/)

Pitfalls

- Modeling aspects:
  - Using constraints that need exponential filtering algorithms.
  - Bad habit of modeling everything with binary constraints.
- Solving your problem with the wrong method (e.g. there exist specialized methods for problems with linear constraints).