Recent Trends in Constraint Optimization and Satisfaction

Nina Narodytska

VMware Research
Introduction

1. PhD, Optimization Research Group, NICTA, Australia
   • Inference algorithms for global constraints (Toby Walsh)

   • Boolean optimization solver
     (Fahiem Bacchus@UofT, Ed Clarke@CMU)

1. Researcher, Samsung Research America
   • Machine learning for computer vision

2. Researcher, VMware Research
   • Applied optimization (for software verification)
   • Interpretable ML
Outline

• **Constraint satisfaction and optimization**
  • Problem modeling
  • Basic principles of constraint solving
  • Learning mechanisms
  • Solvers landscape

• **Solver independent modelling**
  • Advantages and disadvantages
Constraint satisfaction
Theory vs Practice
Theory vs Practice

• Hard from theoretical point of view (NP-hard, P-Space)

• Efficient in practice in many application domains
Theory vs Practice

• Hard from theoretical point of view (NP-hard, P-Space)

• Efficiently solved in practice in many application domains

• Size of the problem is not a good measure of practical hardness
Theory vs Practice

• Small random problems can be very hard for SAT/BDD based techniques (< 100 variables)

• Very large industrial **structured** problems can be efficiently solved (> 100 000 variables)!
Schematic workflow
Workflow

Problem (NL)
Workflow

Problem (NL)

Modeling

Model
Workflow

Problem (NL)

Modeling

Formal Model

Encoding

Model
Workflow

1. **Problem (NL)**
2. **Modeling**
3. **Formal Model**
4. **Data**
5. **Encoding**
6. **Model**

The workflow involves transforming a natural language problem into a formal model through modeling, followed by encoding data into the formal model.
Workflow

- Problem (NL)
- Data
- Encoding
- Formal Model
- Solver model
- Solver
- Encoding
- Model
- Modeling
Workflow

Problem (NL) → Encoding → Formal Model → Encoding → Solver model → Solver → Solution
Workflow

Problem (NL) → Formal Model

Data → Solver model

Modeling

Solver → Solution
Overview

Problem (NL)

Modeling

Data

Formal Model

Solver model

Solver

Solution

Use

Model

Encoding

Encoding
Overview

Modeling

Solving

Use
Bandwidth Allocation Problem (running example)
Bandwidth Allocation Problem

Virtual network

Physical network
Bandwidth Allocation Problem

Virtual network

Physical network
Bandwidth Allocation Problem

Virtual network

Physical network
Bandwidth Allocation Problem

Virtual network

Physical network

Mapping
Modeling
Modeling
Modeling

Logical constraints:
Modeling

Logical constraints:
- each VM is mapped to a host server
Modeling

Logical constraints:
• each VM is mapped to a host server
• for each link between VMs, there is a routing path between the corresponding host servers
Modeling

Logical constraints:
- each VM is mapped to a host server
- for each link between VMs, there is a routing path between the corresponding host servers
- capacity constraints on servers
Modeling

**Logical constraints:**

- each VM is mapped to a host server
- for each link between VMs, there is a routing path between the corresponding host servers
- capacity constraints on servers
- capacity constraints on links
Workflow

Problem (NL)

Data

Modeling

Model
Workflow

Problem (NL) → Data → Formal Model → Model

Encoding
Problem modeling

Solvers modeling language
Problem modeling

Solvers modeling language
Problem modeling

Solvers modeling language

SAT
(T/F)

\((x_1 \lor \neg x_2) \land (x_1 \lor \neg x_3) \land \ldots\)
Problem modeling

Solvers modeling language

CSP

SAT
\[(x_1 \lor \neg x_2) \land (x_1 \lor \neg x_3) \land \ldots\]

Inference

Search
Problem modeling

Solvers modeling language

CSP

MIP (Int/Real)
(2x_1 + x_2 \geq 1) \land 
(5x_1 + 4x_2 \leq 4) \land 
\ldots

SAT (T/F)
(x_1 \lor \neg x_2) \land 
(x_1 \lor \neg x_3) \land 
\ldots

Inference

Search
Problem modeling

Solvers modeling language

CSP

SMT
(Int/Real/Theory)

MIP
(Int/Real)

\[(2x_1 + x_2 \geq 1) \land (5x_1 + 4x_2 \leq 4) \land \ldots\]

SAT
(T/F)

\[(x_1 \lor \neg x_2) \land (x_1 \lor \neg x_3) \land \ldots\]

Inference

Search
Problem modeling

Solvers modeling language

CSP

SMT
 (Int/Real/Theory)

MIP
 (Int/Real)
\[(2x_1 + x_2 \geq 1) \land (5x_1 + 4x_2 \leq 4) \land \ldots\]

SAT
 (T/F)
\[(x_1 \lor \neg x_2) \land (x_1 \lor \neg x_3) \land \ldots\]

Fastest black-box solvers

Inference

Search
Problem modeling

Solvers modeling language

Fast solvers (for verification)

CSP

SMT (Int/Real/Theory)

MIP (Int/Real)

SAT (T/F)

\[(2x_1 + x_2 \geq 1) \land (5x_1 + 4x_2 \leq 4) \land \ldots\]

\[(x_1 \lor \neg x_2) \land (x_1 \lor \neg x_3) \land \ldots\]
Problem modeling

Solvers modeling language

Fast solvers for highly structured problems

CSP

SMT
(Int/Real/Theory)

MIP
(Int/Real)

SAT
(T/F)

Inference

Search

$(2x_1 + x_2 \geq 1) \land (5x_1 + 4x_2 \leq 4) \land \ldots$

$(x_1 \lor \neg x_2) \land (x_1 \lor \neg x_3) \land \ldots$
Problem modeling

Solvers modeling language

- CSP
- SMT (Int/Real/Theory)
- MIP (Int/Real)
- SAT (T/F)

Inference

Search

\[(x_1 \lor \neg x_2) \land (2x_1 + x_2 \geq 1) \land (5x_1 + 4x_2 \leq 4) \land \ldots\]

\[(x_1 \lor \neg x_3) \land \ldots\]
SAT solvers
SAT solvers

Consists of a set of Boolean variables and clauses

\[ x_1, x_2, x_3 \]
SAT solvers

Consists of a set of Boolean variables and clauses

\[ x_1, x_2, x_3 \quad \text{\text{\textcolor{Green}{T}}} \quad \text{\text{\textcolor{Red}{F}}} \quad \neg x_i \]

\[ C_1 = (x_1) \quad C_3 = (x_1 \lor x_2) \]

\[ C_2 = (x_2) \quad C_4 = (\neg x_1 \lor \neg x_3) \]
SAT solvers

Consists of a set of Boolean variables and clauses

\[ x_1, x_2, x_3, T, F, \neg x_i \]

Goal: find an assignment that satisfies all clauses

\[
\begin{align*}
C_1 &= (x_1) & C_3 &= (x_1 \lor x_2) \\
C_2 &= (x_2) & C_4 &= (\neg x_1 \lor \neg x_3)
\end{align*}
\]
SAT solvers

Consists of a set of Boolean variables and clauses

\[ x_1, x_2, x_3 \]

\[ C_1 = (x_1) \quad C_3 = (x_1 \lor x_2) \]

\[ C_2 = (x_2) \quad C_4 = (\neg x_1 \lor \neg x_3) \]
Bandwidth Allocation Problem
Bandwidth Allocation Problem

∀v ∈ VM, ∀s ∈ Server \( X(v, s) \in \{0, 1\} \)

\( X(v, s) = 1 \) iff \( v \) is hosted in \( s \)
Bandwidth Allocation Problem

\( \forall v \in \text{VM}, \forall s \in \text{Server} \) \( X(v, s) \in \{0, 1\} \)

\( X(v, s) = 1 \) iff \( v \) is hosted in \( s \)

(1) each VM is mapped to a host server

\[ \wedge_{v \in \text{VM}} (\sum_{s \in \text{Servers}} X(v, s) = 1) \]
Bandwidth Allocation Problem

\[ \forall v \in VM, \forall s \in Server \ X(v, s) \in \{0, 1\} \]
\[ X(v, s) = 1 \text{ iff } v \text{ is hosted in } s \]

(1) each VM is mapped to a host server

\[ \bigwedge_{v \in VM} \left( \sum_{s \in Servers} X(v, s) = 1 \right) \]

(3) capacity constraints on servers

\[ \bigwedge_{s \in Servers} \left( \sum_{v \in VM} X(v, s) \leq \text{capacity}(s) \right) \]
SAT solvers

Complete search (CDCL search)

• finds a solution, otherwise
• guarantees that there are no solutions

Incomplete search (local search)

• finds a solution, otherwise
• no guarantees that there are no solutions
Problem modeling

CSP

Inference

Search

Solvers modeling language

CSP

SMT
(Int/Real/Theory)

MIP
(Int/Real)

SAT
(T/F)

\[(2x_1 + x_2 \geq 1) \land (5x_1 + 4x_2 \leq 4) \land \ldots\]

\[(x_1 \lor \neg x_2) \land (x_1 \lor \neg x_3) \land \ldots\]
CSP solvers
CSP solvers

Consists of a set of integer (or set) variables and constraints

\[ x_1, x_2, x_3 \]
CSP solvers

Consists of a set of integer (or set) variables and constraints

\[ x_1, x_2, x_3 \]

\[ \text{AllDifferent}(x_1, x_2, \ldots, x_n) \]

\[ \text{Regular}(x_1, x_2, \ldots, x_n, A) \]
CSP solvers

Consists of a set of integer (or set) variables and constraints

\[ x_1, x_2, x_3 \]

Goal: find an assignment that satisfies all constraints

\[ \text{AllDifferent}(x_1, x_2, \ldots, x_n) \]

\[ \text{Regular}(x_1, x_2, \ldots, x_n, A) \]
Bandwidth Allocation Problem
Bandwidth Allocation Problem

\[ \forall v \in VM, X(v) \in \{1, 2, 3\} \]

\[ X(v) = s \text{ iff } v \text{ is hosted in } s \]
Bandwidth Allocation Problem

\[ \forall v \in \text{VM}, X(v) \in \{1, 2, 3\} \]

\[ X(v) = s \text{ iff } v \text{ is hosted in } s \]

(1) each VM is mapped to a host server

No need for a constraint
Bandwidth Allocation Problem

∀v ∈ VM, X(v) ∈ {1, 2, 3}

X(v) = s iff v is hosted in s

(1) each VM is mapped to a host server

No need for a constraint

(3) capacity constraints on servers

GlobalCardConstraint[(X₁, . . . , Xₙ), [capacity(s₁), . . . , capacity(sₘ)]]
Which solver to use?
Which solver to use?

It depends!
Which solver to use?

Understand your problem (under-constrained, over-constrained)
• under-constrained are usually easy to solve by incomplete search
• over-constrained most likely have no solutions
Which solver to use?

• Start with CP model. Use the simplest model possible. Most likely it will be slow.
Which solver to use?

• Start with CP model. Use the simplest model possible. Most likely it will be slow.

• Take advantage of domain specific information
  Remove model symmetry, problem decomposition, heuristics
Which solver to use?

• Start with CP model. Use the simplest model possible. Most likely it will be slow.

• Take advantage of the domain specific information
  Remove model symmetry, problem decomposition, heuristics

• Avoid using complicated variables, e.g. set variables
  It is very hard to reason about them efficiently
Which solver to use?

• Start with CP model. Use the simplest model possible. Most likely it will be slow.

• Take advantage of the domain specific information
  Remove model symmetry, problem decomposition, heuristics

• Avoid using complicated variables, e.g. set variables
  It is very hard to reason about them efficiently

• Relax constraints (e.g. use soft constraints instead of hard constraints)
Backtracking search

\[ x_1 = 1 \]
Backtracking search

\[ X_1 = 1 \]

\[ X_4 = 3 \]
Backtracking search

\[ X_1 = 1 \]

\[ X_4 = 3 \]

\[ X_5 = 1 \]
Backtracking search

$X_1 = 1$

$X_4 = 3$

$X_5 = 1$

$X_5 = 2$
Backtracking search

X₁ = 1
X₁ = 2
X₄ = 3
X₅ = 1
Backtracking search

\[ X_1 = 1 \]
\[ X_1 = 2 \]
\[ X_4 = 3 \]
\[ X_5 = 1 \]

Virtual network

Physical network
Backtracking search

Key to the success of modern solvers is learning from failures
Learning mechanism

$X_1 = 1$

$X_4 = 3$

$X_5 = 1$
Learning mechanism

\[ X_1 = 1 \]

\[ X_4 = 3 \]

\[ X_5 = 1 \]

\[ \text{NOT } (X_4 = 3 \ \text{AND} \ X_5 = 1) \]
Learning mechanism

\[ X_1 = 1 \]
\[ X_4 = 3 \]
\[ X_5 = 1 \]

\[ X_1 = 2 \]

\[ \text{NOT (} X_4 = 3 \text{ AND } X_5 = 1 \text{)} \]
Learning mechanism

\[ \text{NOT} \ (X_4 = 3 \ \text{AND} \ X_5 = 1) \]
CP solvers best learning model

High level model (CP)
CP solvers best learning model

High level model (CP)

Lower level model (SAT)
CP solvers best learning model

High level model (CP) ≠ Lower level model (SAT)
CP solvers best learning model

High level model (CP)

Lower level model (SAT)
CP solvers best learning model

AllDifferent(X,Y,Z)

X, Y ∈ \{1,2\}
Z ∈ \{1,2,3\}
CP solvers best learning model

AllDifferent(X,Y,Z)

X, Y ∈ \{1,2\}
Z ∈ \{1,2,3\}

SAT
CP solvers best learning model

If \( X, Y \in \{1,2\} \) then \( Z \not\in \{1,2\} \)

AllDifferent(\(X,Y,Z\))

\( X, Y \in \{1,2\} \)

\( Z \in \{1,2,3\} \)
CP solvers best learning model

\[ \text{AllDifferent}(X, Y, Z) \]

\[ X, Y \in \{1, 2\} \]
\[ Z \in \{1, 2, 3\} \]

If \( X, Y \in \{1, 2\} \) then \( Z \not\in \{1, 2\} \)

Designed for each constraint
CP solvers best learning model
CP solvers best learning model

CP

Z = 1

SAT
CP solvers best learning model

Lazy clause generation
How solvers learn

CSP  SMT  MIP  SAT
How solvers learn

- CSP
- SMT
- MIP
- SAT

Simpler modeling language makes it easier to define an efficient learning scheme
How solvers learn

- SAT: learn clauses
- MIP: learn linear constraints
- CP: there is no mechanism to learn global constraints,
- CP/SAT hybrid solvers extract explanations from global constraints and learn clauses

Simpler modeling language makes it easier to define an efficient learning scheme
How solvers learn

- SAT: learn clauses
- MIP: learn linear constraints
- CP: there is no mechanism to learn global constraints,
- CP/SAT hybrid solvers extract explanations from global constraints and learn clauses

Simpler modeling language makes it easier to define an efficient learning scheme
Use of the technology

- SAT and MIP are the fastest generic complete search solvers (used in industrial applications)

- Learning-based CP solvers are good alternatives if the problem has rich structure or the problem is tight.
What if it does not work

- Performance debugging is a challenge
- Design a simple greedy search
  - Greedy algorithm, LS algorithm are usually domain specific.
    - hint for powerful heuristics
  - Understand what are good heuristics for your problem

- Guide CP solver using the same heuristic
  - E.g. alter branching heuristics
Solvers landscape

- OR-Tools LCG (Google)
- Chuffed
- Choco
Solvers landscape

- CSP
  - OR-Tools LCG (Google)
  - Chuff
  - Choco

- SMT
  - Z3 (MSR)
  - CVC4 (Stanford, Iowa)

- MIP

- SAT
Solvers landscape

- **CSP**
  - OR-Tools LCG (Google)
  - Chuff
  - Choco

- **SMT**
  - Z3 (MSR)
  - CVC4 (Stanford, Iowa)

- **MIP**
  - CPLEX
  - gurobi
  - SCIP
  - OR-Tools LCG

- **SAT**
Solvers landscape

- **CSP**
  - OR-Tools LCG (Google)
  - Chuff
  - Choco

- **SMT**
  - Z3 (MSR)
  - CVC4 (Stanford, Iowa)

- **MIP**
  - CPLEX
  - gurobi
  - SCIP
  - OR-Tools LCG

- **SAT**
  - Lingeling
  - Glucose
Solver independent modeling

Solvers modeling language

CSP

SMT
(Int/Real/Theory)

MIP
(Int/Real)

SAT
(T/F)

\[(2x_1 + x_2 \geq 1) \land (5x_1 + 4x_2 \leq 4) \land \ldots \]

\[(x_1 \lor \neg x_2) \land (x_1 \lor \neg x_3) \land \ldots \]
Solver independent modeling

Solvers modeling language

Minizinc

CSP

SMT
(Int/Real/Theory)

MIP
(Int/Real)

(2x_1 + x_2 \geq 1) \land 
(5x_1 + 4x_2 \leq 4) \land 
\ldots

SAT
(T/F)

(x_1 \lor \neg x_2) \land 
(x_1 \lor \neg x_3) \land 
\ldots
Solver independent modeling

- Great tool for problem specification
- Allows passing domain specific knowledge to the solver
- Do not mix different classes of variables, e.g. integer and set variables unless it is really necessary
Is it a magic tool?

No, for any solver, one can find a small problem on which it never terminates, e.g. a pigeon hole problem for SAT.
Should I use them?

Yes, these are the best technologies out there.

An alternative would be to craft a new greedy search-based solver for each small variation of the problem.
Thanks!