UNIVERSITY OF HELSINKI
FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE
PHD THESIS

IMPLICIT HITTING SET ALGORITHMS FOR CONSTRAINT OPTIMIZATION

PAUL SAIKKO

What?

Implicit Hitting Set Algorithms for Constraint Optimization

What?

Implicit Hitting Set Algorithms for Constraint Optimization

Algorithms

Focus of the thesis

Design, implement, and evaluate *algorithms*.

Algorithm

A sequence of instructions that solves a class of problems.

Problems

Decision

Are there any solutions?

Optimization

Find a best solution, for example:

- Fastest route
- Most efficient schedule
- Most probable cause
- etc...

Complexity

This thesis: Optimization when the corresponding decision problem is NP-complete (or harder)

Problems

Decision

Are there any solutions?

Optimization

Find a best solution, for example:

- Fastest route
- Most efficient schedule
- Most probable cause
- etc...

Complexity

This thesis: Optimization when the corresponding decision problem is NP-complete (or harder)

Problems

Decision

Are there any solutions?

Optimization

Find a best solution, for example:

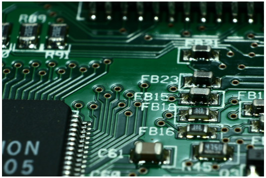
- Fastest route
- Most efficient schedule
- Most probable cause
- etc...

Complexity

This thesis: Optimization when the corresponding decision problem is NP-complete (or harder)

Applications

- Route planning:
  e.g. autonomous vehicles in warehouses
- Scheduling:
  e.g. assigning terminals, timeslots, or runways for air traffic
- Hardware design:
  e.g. chip layout optimization



Declarative Approach

Alternative to special-purpose algorithms:

1. **Model** problem in a constraint language as a *set of constraints*
2. **Solve** using a generic algorithm (solver) for that constraint language
3. **Reconstruct** a solution for the original problem

Benefits

- Easy to refine and extend problem definition
- Solver development benefits many different problem domains

Declarative Approach

Alternative to special-purpose algorithms:

1. **Model** problem in a constraint language as a *set of constraints*
2. **Solve** using a generic algorithm (solver) for that constraint language
3. **Reconstruct** a solution for the original problem

Benefits

- Easy to refine and extend problem definition
- Solver development benefits many different problem domains

Declarative Approach

Alternative to special-purpose algorithms:

1. **Model** problem in a constraint language as a *set of constraints*
2. **Solve** using a generic algorithm (solver) for that constraint language
3. **Reconstruct** a solution for the original problem

Benefits

- Easy to refine and extend problem definition
- Solver development benefits many different problem domains

Declarative Approach

Alternative to special-purpose algorithms:

1. **Model** problem in a constraint language as a *set of constraints*
2. **Solve** using a generic algorithm (solver) for that constraint language
3. **Reconstruct** a solution for the original problem

Benefits

- Easy to refine and extend problem definition
- Solver development benefits many different problem domains

Constraint Languages

Many approaches to model and solve constrained optimization problems:

- Integer and linear programming (ILP / LP)
- Finite-domain constraint satisfaction/optimization (CP)
- Boolean satisfiability (SAT)
- Satisfiability modulo theories (SMT)
- Maximum satisfiability (MaxSAT)
- Answer set programming (ASP)
- etc ...

Constraint Languages

Many approaches to model and solve constrained optimization problems:

- **Integer and linear programming (ILP / LP)**
- Finite-domain constraint satisfaction/optimization (CP)
- **Boolean satisfiability (SAT)**
- Satisfiability modulo theories (SMT)
- **Maximum satisfiability (MaxSAT)**
- **Answer set programming (ASP)**
- etc ...

How?

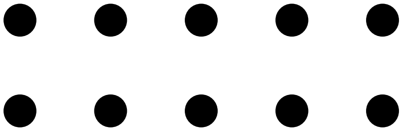
Implicit Hitting Set Algorithms for Constraint Optimization

How?

Implicit Hitting Set Algorithms for Constraint Optimization

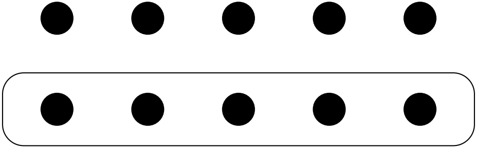
Hitting Sets

- Set of elements: circles



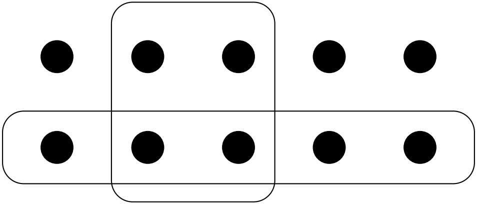
Hitting Sets

- Set of elements: circles
- Subsets to hit: rounded rectangles



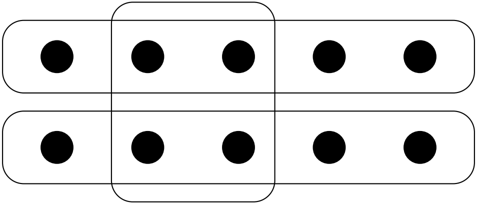
Hitting Sets

- Set of elements: circles
- Subsets to hit: rounded rectangles



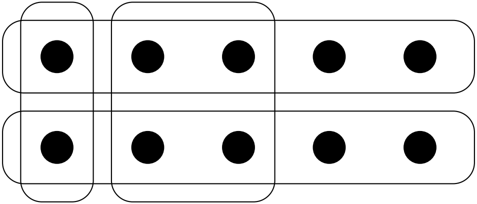
Hitting Sets

- Set of elements: circles
- Subsets to hit: rounded rectangles



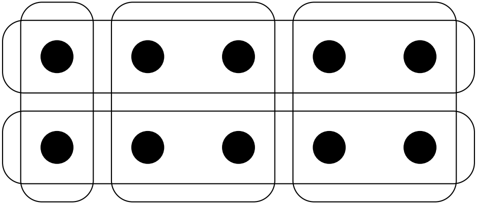
Hitting Sets

- Set of elements: circles
- Subsets to hit: rounded rectangles



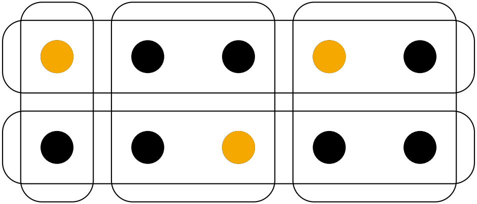
Hitting Sets

- Set of elements: circles
- Subsets to hit: rounded rectangles



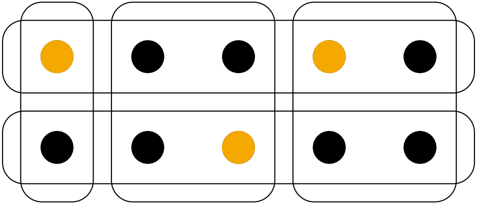
Hitting Sets

- Set of elements: circles
- Subsets to hit: rounded rectangles
- A hitting set: orange circles



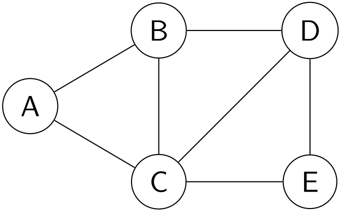
Hitting Sets

- Set of elements: circles
- Subsets to hit: rounded rectangles
- A hitting set: orange circles



NP-hard problem: Find smallest or minimum-cost hitting set

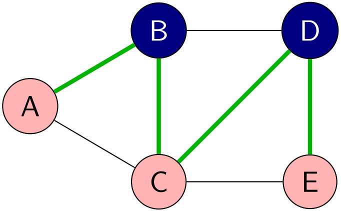
Optimizing with Hitting Sets



Max Cut

Partition vertices into two sets, maximizing the number of edges between them.

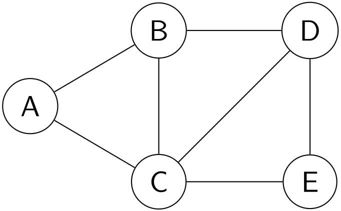
Optimizing with Hitting Sets



Max Cut

Partition vertices into two sets, maximizing the number of edges between them.

Optimizing with Hitting Sets



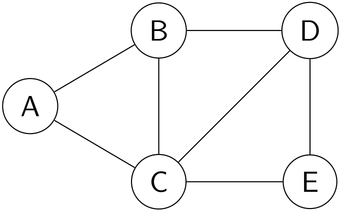
Max Cut

Partition vertices into two sets, maximizing the number of edges between them.

Constraints

For each edge: endpoints in different partitions (e.g. "$A \neq B$")

Optimizing with Hitting Sets



Max Cut

Partition vertices into two sets, maximizing the number of edges between them.

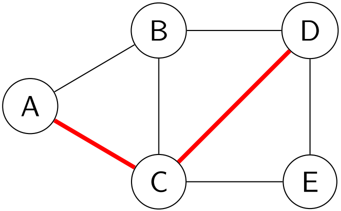
Constraints

For each edge: endpoints in different partitions (e.g. "$A \neq B$")

Cores

$A \neq B$ or $B \neq C$ or $A \neq C$,
$B \neq C$ or $C \neq D$ or $B \neq D$,
$C \neq D$ or $D \neq E$ or $C \neq E$,
$A \neq B$ or $B \neq D$ or $D \neq E$ or $C \neq E$ or $A \neq C$

Optimizing with Hitting Sets



Max Cut

Partition vertices into two sets, maximizing the number of edges between them.

Constraints

For each edge: endpoints in different partitions (e.g. "$A \neq B$")

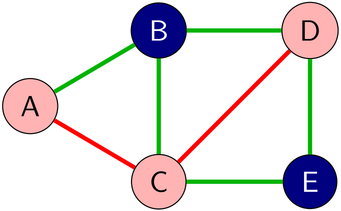
Cores

$A \neq B$ or $B \neq C$ or $A \neq C$,
$B \neq C$ or $C \neq D$ or $B \neq D$,
$C \neq D$ or $D \neq E$ or $C \neq E$,
$A \neq B$ or $B \neq D$ or $D \neq E$ or $C \neq E$ or $A \neq C$

Hitting Set

$A \neq C$ and $C \neq D$

Optimizing with Hitting Sets



Max Cut

Partition vertices into two sets, maximizing the number of edges between them.

Constraints

For each edge: endpoints in different partitions (e.g. "$A \neq B$")

Cores

$A \neq B$ or $B \neq C$ or $A \neq C$,
$B \neq C$ or $C \neq D$ or $B \neq D$,
$C \neq D$ or $D \neq E$ or $C \neq E$,
$A \neq B$ or $B \neq D$ or $D \neq E$ or $C \neq E$ or $A \neq C$

Hitting Set

$A \neq C$ and $C \neq D$

Generalizing

- **Core**: A subset of constraints that cannot be simultaneously satisfied
- A **solution** must leave out a part of each core
- Unsatisfied constraints form a **hitting set** of the set of all cores
- If the solution is optimal, this is a **minimum hitting set**

Generalizing

- **Core**: A subset of constraints that cannot be simultaneously satisfied
- A **solution** must leave out a part of each core
- Unsatisfied constraints form a **hitting set** of the set of all cores
- If the solution is optimal, this is a **minimum hitting set**

Generalizing

- **Core**: A subset of constraints that cannot be simultaneously satisfied
- A **solution** must leave out a part of each core
- Unsatisfied constraints form a **hitting set** of the set of all cores
- If the solution is optimal, this is a **minimum hitting set**

Generalizing

- **Core**: A subset of constraints that cannot be simultaneously satisfied
- A **solution** must leave out a part of each core
- Unsatisfied constraints form a **hitting set** of the set of all cores
- If the solution is optimal, this is a **minimum hitting set**

Developing a Viable Approach

- Finding even one core is often a complex task.
  (We need to solve NP-complete problems)
- Problem can have exponential number of cores.
  (Even in the simple Max Cut example)
- Can we find a hitting set of all cores *without knowing all cores*?

Developing a Viable Approach

- Finding even one core is often a complex task.
  (We need to solve NP-complete problems)

- Problem can have exponential number of cores.
  (Even in the simple Max Cut example)

- Can we find a hitting set of all cores *without knowing all cores*?

Developing a Viable Approach

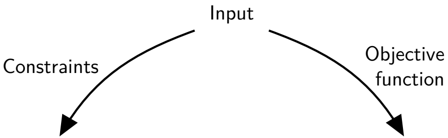
- Finding even one core is often a complex task.
  (We need to solve NP-complete problems)
- Problem can have exponential number of cores.
  (Even in the simple Max Cut example)
- Can we find a hitting set of all cores *without knowing all cores*?

Implicit Hitting Set Algorithms

- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem

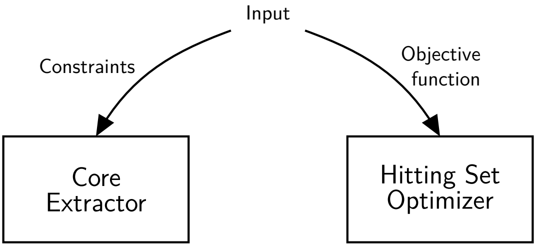
Implicit Hitting Set Algorithms

- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem



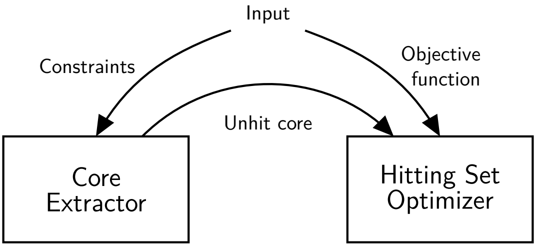
Implicit Hitting Set Algorithms

- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem



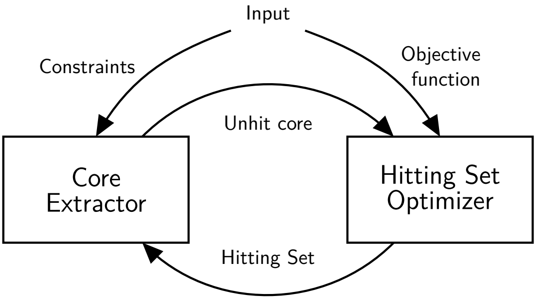
Implicit Hitting Set Algorithms

- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem



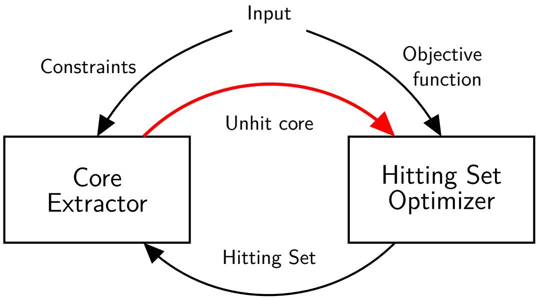
Implicit Hitting Set Algorithms

- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem



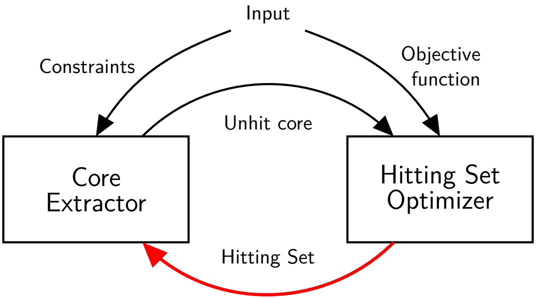
Implicit Hitting Set Algorithms

- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem



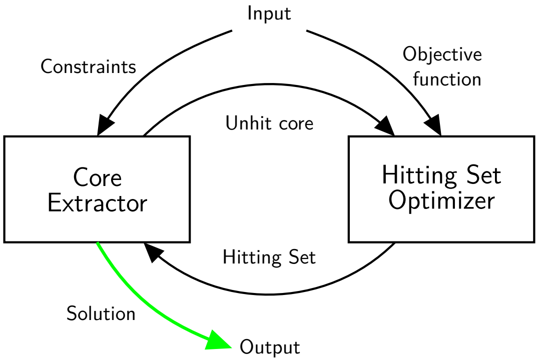
Implicit Hitting Set Algorithms

- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem



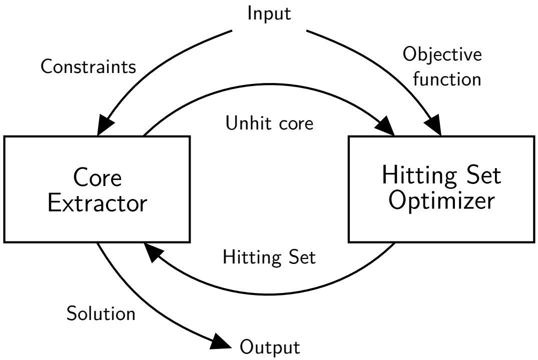
Implicit Hitting Set Algorithms

- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem

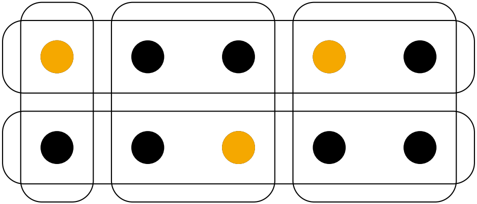


Implicit Hitting Set Algorithms

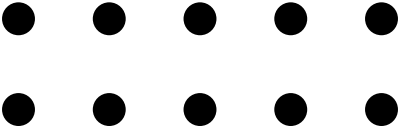
- Alternate between finding cores and computing hitting sets
- Solve hitting set problems with e.g. integer programming
- Find cores with a solver for the corresponding decision problem



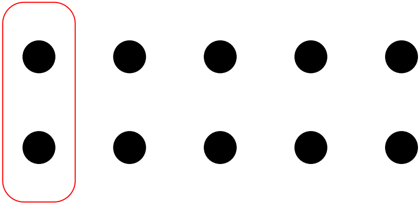
Implicit Hitting Sets



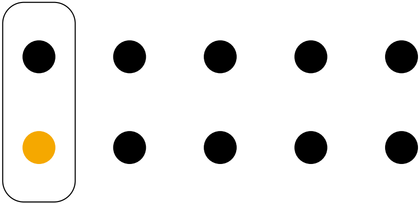
Implicit Hitting Sets



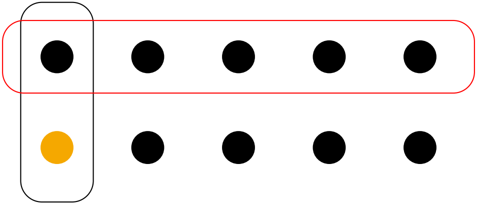
Implicit Hitting Sets



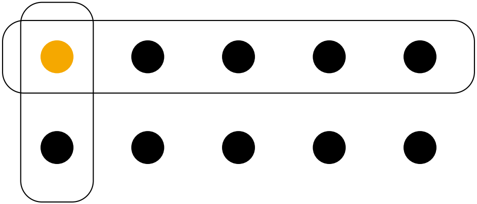
Implicit Hitting Sets



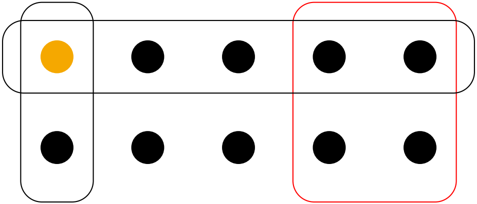
Implicit Hitting Sets



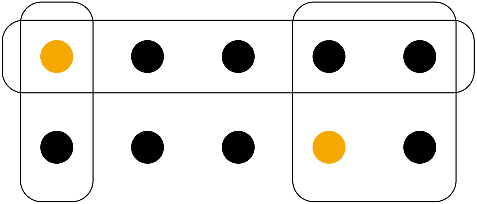
Implicit Hitting Sets



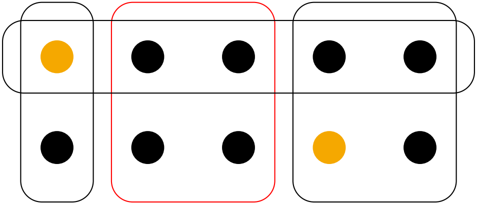
Implicit Hitting Sets



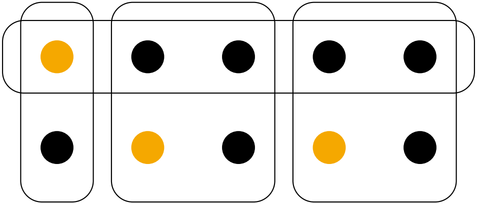
Implicit Hitting Sets



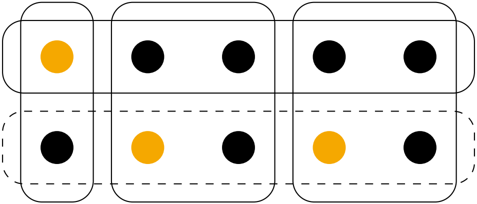
Implicit Hitting Sets



Implicit Hitting Sets



Implicit Hitting Sets



Contributions

Instantiations of the IHS algorithms to 4 constraint optimization domains:

- MaxSAT (Papers I, II)
- Causal structure learning (Paper III)
- Abductive reasoning (Paper IV)
- Answer set optimization (Paper V)

Contributions

Instantiations of the IHS algorithms to 4 constraint optimization domains:

- MaxSAT (Papers I, II)
- Causal structure learning (Paper III)
- Abductive reasoning (Paper IV)
- Answer set optimization (Paper V)

Contributions

Instantiations of the IHS algorithms to 4 constraint optimization domains:

- MaxSAT (Papers I, II)
- Causal structure learning (Paper III)
- Abductive reasoning (Paper IV)
- Answer set optimization (Paper V)

Contributions

Instantiations of the IHS algorithms to 4 constraint optimization domains:

- MaxSAT (Papers I, II)
- Causal structure learning (Paper III)
- Abductive reasoning (Paper IV)
- Answer set optimization (Paper V)

IHS for Constraint Languages

General purpose constraint optimization languages

- Applications in various domains

MaxSAT

- Constraints: CNF clauses "$l_1 \vee \cdots \vee l_n$"
- Core extractor: SAT solver (NP-complete)

Answer set optimization

- Constraints: inference rules "$p_1 \vee \cdots \vee p_m \leftarrow l_1 \wedge \cdots \wedge l_n$"
- Core extractor: ASP solver (Σ_2^P-complete)

IHS for Constraint Languages

General purpose constraint optimization languages

- Applications in various domains

MaxSAT

- Constraints: CNF clauses "$l_1 \vee \cdots \vee l_n$"
- Core extractor: SAT solver (NP-complete)

Answer set optimization

- Constraints: inference rules "$p_1 \vee \cdots \vee p_m \leftarrow l_1 \wedge \cdots \wedge l_n$"
- Core extractor: ASP solver (Σ_2^P-complete)

IHS for Constraint Languages

General purpose constraint optimization languages

- Applications in various domains

MaxSAT

- Constraints: CNF clauses "$l_1 \vee \cdots \vee l_n$"
- Core extractor: SAT solver (NP-complete)

Answer set optimization

- Constraints: inference rules "$p_1 \vee \cdots \vee p_m \leftarrow l_1 \wedge \cdots \wedge l_n$"
- Core extractor: ASP solver (Σ_2^P-complete)

Causal Structure Learning

Problem setting

1. Observations of random variables
2. (In)dependencies from statistical tests
3. Constraint encoding of d-separation conditions
4. Computing *globally optimal* causal structure

X_1	X_2	X_3
0.1	-0.34	0.8
0.22	-0.4	-0.1
$\vdots$	$\vdots$	$\vdots$

DATA

$\Rightarrow$

k	$w(k)$
$X_1 \perp\!\!\!\perp X_3$	3.29
$X_1 \perp\!\!\!\perp X_3 \mid X_2$	3.73
$X_2 \not\perp\!\!\!\perp X_3$	23.4
$X_2 \not\perp\!\!\!\perp X_3 \mid X_1$	21.2
$X_1 \not\perp\!\!\!\perp X_2$	15.8
$X_1 \not\perp\!\!\!\perp X_2 \mid X_3$	10.11

(IN)DEPENDENCIES

$\Rightarrow$ IHS optimization $\Rightarrow$

CAUSAL GRAPH STRUCTURE

Causal Structure Learning

Problem setting

1. Observations of random variables
2. (In)dependencies from statistical tests
3. Constraint encoding of d-separation conditions
4. Computing *globally optimal* causal structure

X_1	X_2	X_3
0.1	-0.34	0.8
0.22	-0.4	-0.1
$\vdots$	$\vdots$	$\vdots$

DATA

$\Rightarrow$

k	$w(k)$
$X_1 \perp\!\!\!\perp X_3$	3.29
$X_1 \perp\!\!\!\perp X_3 \mid X_2$	3.73
$X_2 \not\perp\!\!\!\perp X_3$	23.4
$X_2 \not\perp\!\!\!\perp X_3 \mid X_1$	21.2
$X_1 \not\perp\!\!\!\perp X_2$	15.8
$X_1 \not\perp\!\!\!\perp X_2 \mid X_3$	10.11

(IN)DEPENDENCIES

$\Rightarrow$ IHS optimization $\Rightarrow$

CAUSAL GRAPH STRUCTURE

Causal Structure Learning

Problem setting

1. Observations of random variables
2. (In)dependencies from statistical tests
3. Constraint encoding of d-separation conditions
4. Computing *globally optimal* causal structure

X_1	X_2	X_3
0.1	-0.34	0.8
0.22	-0.4	-0.1
$\vdots$	$\vdots$	$\vdots$

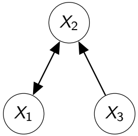
DATA

$\Rightarrow$

k	$w(k)$
$X_1 \perp\!\!\!\perp X_3$	3.29
$X_1 \perp\!\!\!\perp X_3 \mid X_2$	3.73
$X_2 \not\perp\!\!\!\perp X_3$	23.4
$X_2 \not\perp\!\!\!\perp X_3 \mid X_1$	21.2
$X_1 \not\perp\!\!\!\perp X_2$	15.8
$X_1 \not\perp\!\!\!\perp X_2 \mid X_3$	10.11

(IN)DEPENDENCIES

$\Rightarrow$ IHS optimization $\Rightarrow$



CAUSAL GRAPH STRUCTURE

Causal Structure Learning

Problem setting

1. Observations of random variables
2. (In)dependencies from statistical tests
3. Constraint encoding of d-separation conditions
4. Computing *globally optimal* causal structure

X_1	X_2	X_3
0.1	-0.34	0.8
0.22	-0.4	-0.1
$\vdots$	$\vdots$	$\vdots$

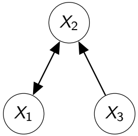
DATA

$\Rightarrow$

k	$w(k)$
$X_1 \perp\!\!\!\perp X_3$	3.29
$X_1 \perp\!\!\!\perp X_3 \mid X_2$	3.73
$X_2 \not\!\perp\!\!\!\perp X_3$	23.4
$X_2 \not\!\perp\!\!\!\perp X_3 \mid X_1$	21.2
$X_1 \not\!\perp\!\!\!\perp X_2$	15.8
$X_1 \not\!\perp\!\!\!\perp X_2 \mid X_3$	10.11

(IN)DEPENDENCIES

$\Rightarrow$ IHS optimization $\Rightarrow$



CAUSAL GRAPH STRUCTURE

Causal Structure Learning

Problem setting

1. Observations of random variables
2. (In)dependencies from statistical tests
3. Constraint encoding of d-separation conditions
4. Computing *globally optimal* causal structure

X_1	X_2	X_3
0.1	-0.34	0.8
0.22	-0.4	-0.1
$\vdots$	$\vdots$	$\vdots$

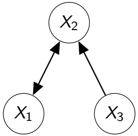
DATA

$\Rightarrow$

k	$w(k)$
$X_1 \perp\!\!\!\perp X_3$	3.29
$X_1 \perp\!\!\!\perp X_3 \mid X_2$	3.73
$X_2 \not\!\perp\!\!\!\perp X_3$	23.4
$X_2 \not\!\perp\!\!\!\perp X_3 \mid X_1$	21.2
$X_1 \not\!\perp\!\!\!\perp X_2$	15.8
$X_1 \not\!\perp\!\!\!\perp X_2 \mid X_3$	10.11

(IN)DEPENDENCIES

$\Rightarrow$ IHS optimization $\Rightarrow$



CAUSAL GRAPH STRUCTURE

Causal Structure Learning

Problem setting

1. Observations of random variables
2. (In)dependencies from statistical tests
3. Constraint encoding of d-separation conditions
4. Computing *globally optimal* causal structure

X_1	X_2	X_3
0.1	-0.34	0.8
0.22	-0.4	-0.1
$\vdots$	$\vdots$	$\vdots$

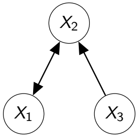
DATA

$\Rightarrow$

k	$w(k)$
$X_1 \perp\!\!\!\perp X_3$	3.29
$X_1 \perp\!\!\!\perp X_3 \mid X_2$	3.73
$X_2 \not\!\perp\!\!\!\perp X_3$	23.4
$X_2 \not\!\perp\!\!\!\perp X_3 \mid X_1$	21.2
$X_1 \not\!\perp\!\!\!\perp X_2$	15.8
$X_1 \not\!\perp\!\!\!\perp X_2 \mid X_3$	10.11

(IN)DEPENDENCIES

$\Rightarrow$ IHS optimization $\Rightarrow$



CAUSAL GRAPH STRUCTURE

Causal Structure Learning

Connection to IHS

$Q \not\perp\!\!\!\perp X$	$Y \not\perp\!\!\!\perp Z \mid X, W$	$X \not\perp\!\!\!\perp Z \mid Y, W$	$Q \perp\!\!\!\perp Z$
$Y \not\perp\!\!\!\perp Z \mid W$	$X \perp\!\!\!\perp Y \mid Z, W$	$X \perp\!\!\!\perp Y \mid W$	$X \not\perp\!\!\!\perp Z \mid W$
$X \not\perp\!\!\!\perp W$	$X \not\perp\!\!\!\perp W \mid Z$	$X \not\perp\!\!\!\perp Y \mid W, Q$	$Q \not\perp\!\!\!\perp Y \mid X$
$W \not\perp\!\!\!\perp Z$	$Q \perp\!\!\!\perp W$	$Y \perp\!\!\!\perp Q \mid W$	$Q \perp\!\!\!\perp Z \mid W$

Causal Structure Learning

Connection to IHS

$Q \not\perp\!\!\!\perp X$	$Y \not\perp\!\!\!\perp Z \mid X, W$	$X \not\perp\!\!\!\perp Z \mid Y, W$	$Q \perp\!\!\!\perp Z$
$Y \not\perp\!\!\!\perp Z \mid W$	$X \perp\!\!\!\perp Y \mid Z, W$	$X \perp\!\!\!\perp Y \mid W$	$X \not\perp\!\!\!\perp Z \mid W$
$X \not\perp\!\!\!\perp W$	$X \not\perp\!\!\!\perp W \mid Z$	$X \not\perp\!\!\!\perp Y \mid W, Q$	$Q \not\perp\!\!\!\perp Y \mid X$
$W \not\perp\!\!\!\perp Z$	$Q \perp\!\!\!\perp W$	$Y \perp\!\!\!\perp Q \mid W$	$Q \perp\!\!\!\perp Z \mid W$

Causal Structure Learning

Connection to IHS

$Q \not\perp\!\!\!\perp X$	$Y \not\perp\!\!\!\perp Z \mid X, W$	$X \not\perp\!\!\!\perp Z \mid Y, W$	$Q \perp\!\!\!\perp Z$
$Y \not\perp\!\!\!\perp Z \mid W$	$X \perp\!\!\!\perp Y \mid Z, W$	$X \perp\!\!\!\perp Y \mid W$	$X \not\perp\!\!\!\perp Z \mid W$
$X \not\perp\!\!\!\perp W$	$X \not\perp\!\!\!\perp W \mid Z$	$X \not\perp\!\!\!\perp Y \mid W, Q$	$Q \not\perp\!\!\!\perp Y \mid X$
$W \not\perp\!\!\!\perp Z$	$Q \perp\!\!\!\perp W$	$Y \perp\!\!\!\perp Q \mid W$	$Q \perp\!\!\!\perp Z \mid W$

Causal Structure Learning

Connection to IHS

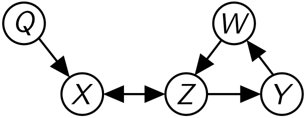
$Q \not\perp\!\!\!\perp X$	$Y \not\perp\!\!\!\perp Z \mid X, W$	$X \not\perp\!\!\!\perp Z \mid Y, W$	$Q \perp\!\!\!\perp Z$
$Y \not\perp\!\!\!\perp Z \mid W$	$X \perp\!\!\!\perp Y \mid Z, W$	$X \perp\!\!\!\perp Y \mid W$	$X \not\perp\!\!\!\perp Z \mid W$
$X \not\perp\!\!\!\perp W$	$X \not\perp\!\!\!\perp W \mid Z$	$X \not\perp\!\!\!\perp Y \mid W, Q$	$Q \not\perp\!\!\!\perp Y \mid X$
$W \not\perp\!\!\!\perp Z$	$Q \perp\!\!\!\perp W$	$Y \perp\!\!\!\perp Q \mid W$	$Q \perp\!\!\!\perp Z \mid W$

Causal Structure Learning

Connection to IHS

$Q \not\perp\!\!\!\perp X$	$Y \not\perp\!\!\!\perp Z \mid X, W$	$X \not\perp\!\!\!\perp Z \mid Y, W$	$Q \perp\!\!\!\perp Z$
$Y \not\perp\!\!\!\perp Z \mid W$	$X \perp\!\!\!\perp Y \mid Z, W$	$X \perp\!\!\!\perp Y \mid W$	$X \not\perp\!\!\!\perp Z \mid W$
$X \not\perp\!\!\!\perp W$	$X \not\perp\!\!\!\perp W \mid Z$	$X \not\perp\!\!\!\perp Y \mid W, Q$	$Q \not\perp\!\!\!\perp Y \mid X$
$W \not\perp\!\!\!\perp Z$	$Q \perp\!\!\!\perp W$	$Y \perp\!\!\!\perp Q \mid W$	$Q \perp\!\!\!\perp Z \mid W$

$$\Downarrow$$



Abductive Reasoning

- Abduction:
  Explanation finding problem
- Optimization:
  What is the simplest explanation?
- Formalization:
  CNF propositional formulas

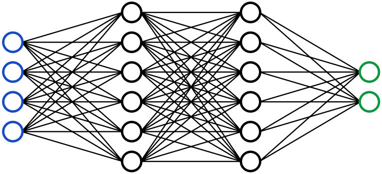
Abductive Reasoning

- Abduction:
  Explanation finding problem
- Optimization:
  What is the simplest explanation?
- Formalization:
  CNF propositional formulas

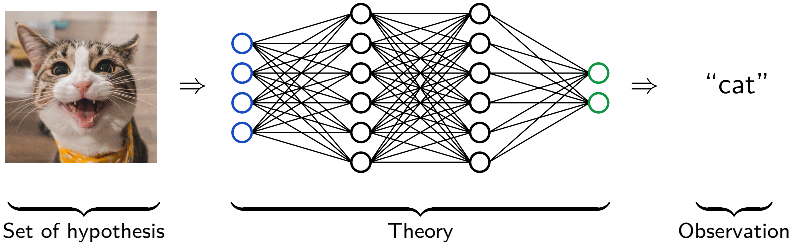
Abductive Reasoning

- Abduction:
  Explanation finding problem
- Optimization:
  What is the simplest explanation?
- Formalization:
  CNF propositional formulas

Abductive Reasoning

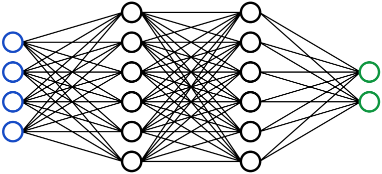
 ⇒  ⇒ "cat"

Abductive Reasoning



Abductive Reasoning

$\Rightarrow$



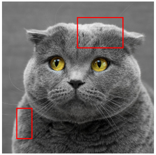
$\Rightarrow$ “bicycle”

Set of hypothesis

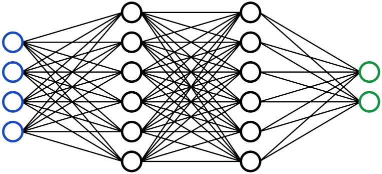
Theory

Observation

Abductive Reasoning



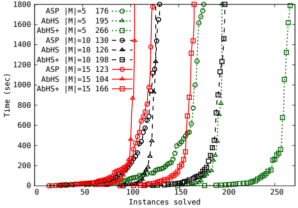
⇒

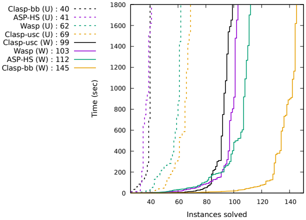


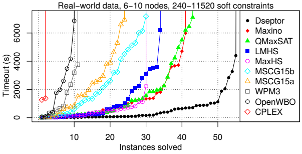
⇒ “bicycle”

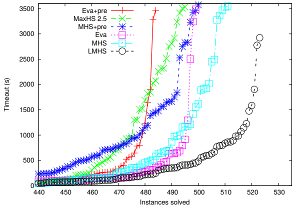
Set of hypothesis — Theory — Observation

Contributions









- Practical implementations of optimization algorithms
- Empirical evaluations show performance
- Open source code

UNIVERSITY OF HELSINKI
FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE
PHD THESIS

IMPLICIT HITTING SET ALGORITHMS FOR CONSTRAINT OPTIMIZATION

PAUL SAIKKO