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)

2. Postdoc. Researcher, Univ. of Toronto and Carnegie Mellon Ur
   - 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

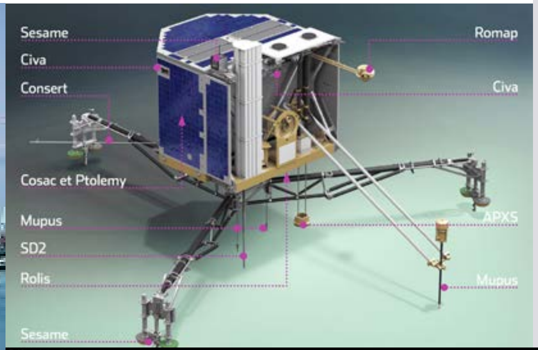
Outline

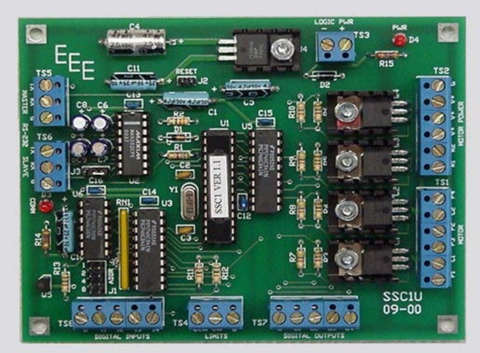
- **Constraint satisfaction and optimization**
  - Problem modeling
  - Basic principles of constraint solving
  - Learning mechanisms
  - Solvers landscape

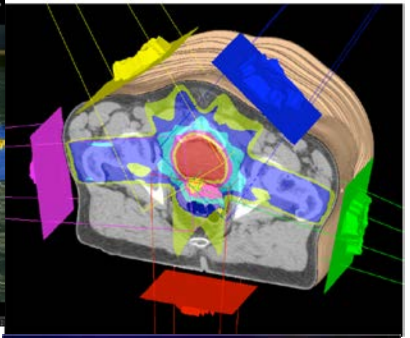
- **Solver independent modelling**
  - Advantages and disadvantages

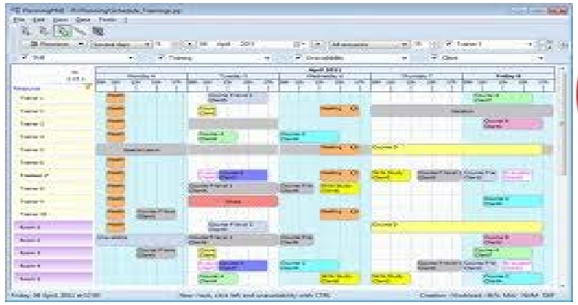
Constraint satisfaction

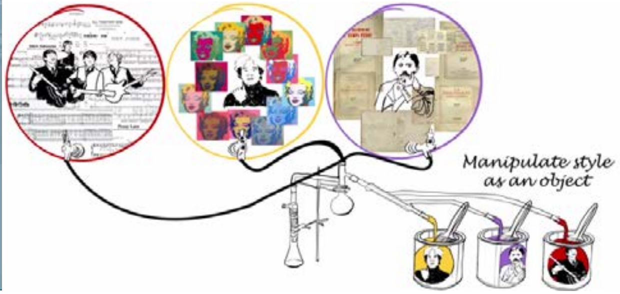
Theory vs Practice











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

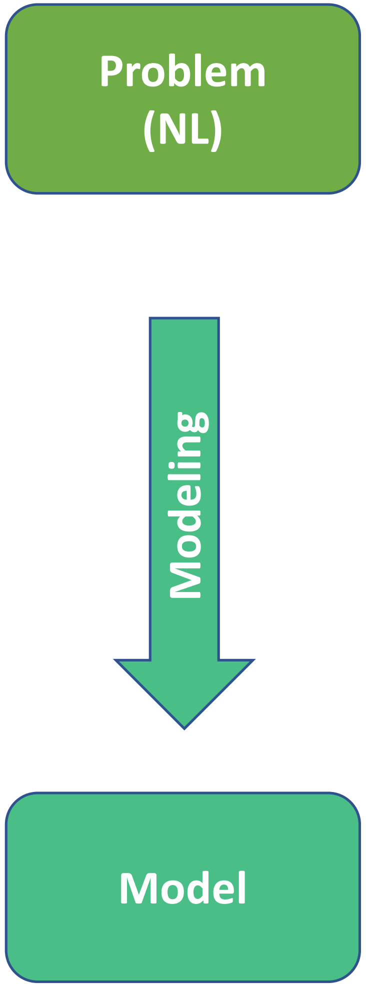
(> 100 000 variables)!

Schematic workflow

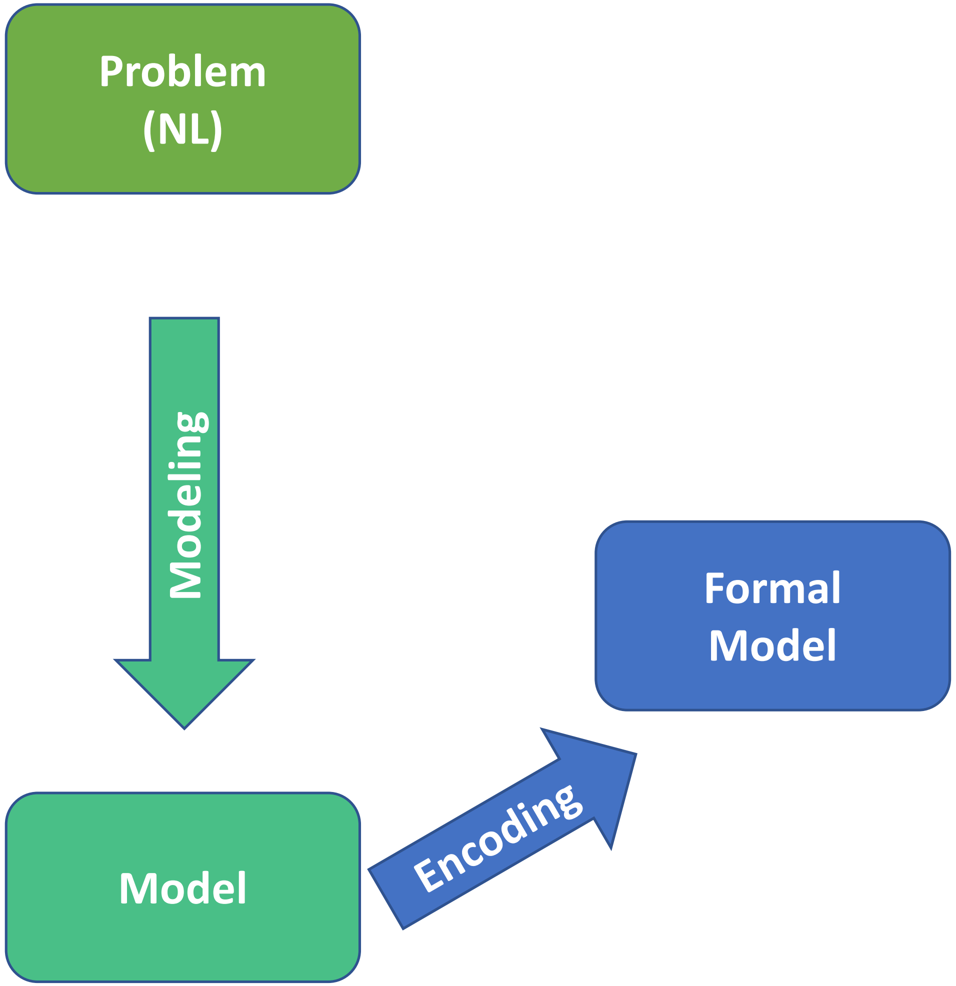
Workflow

**Problem
(NL)**

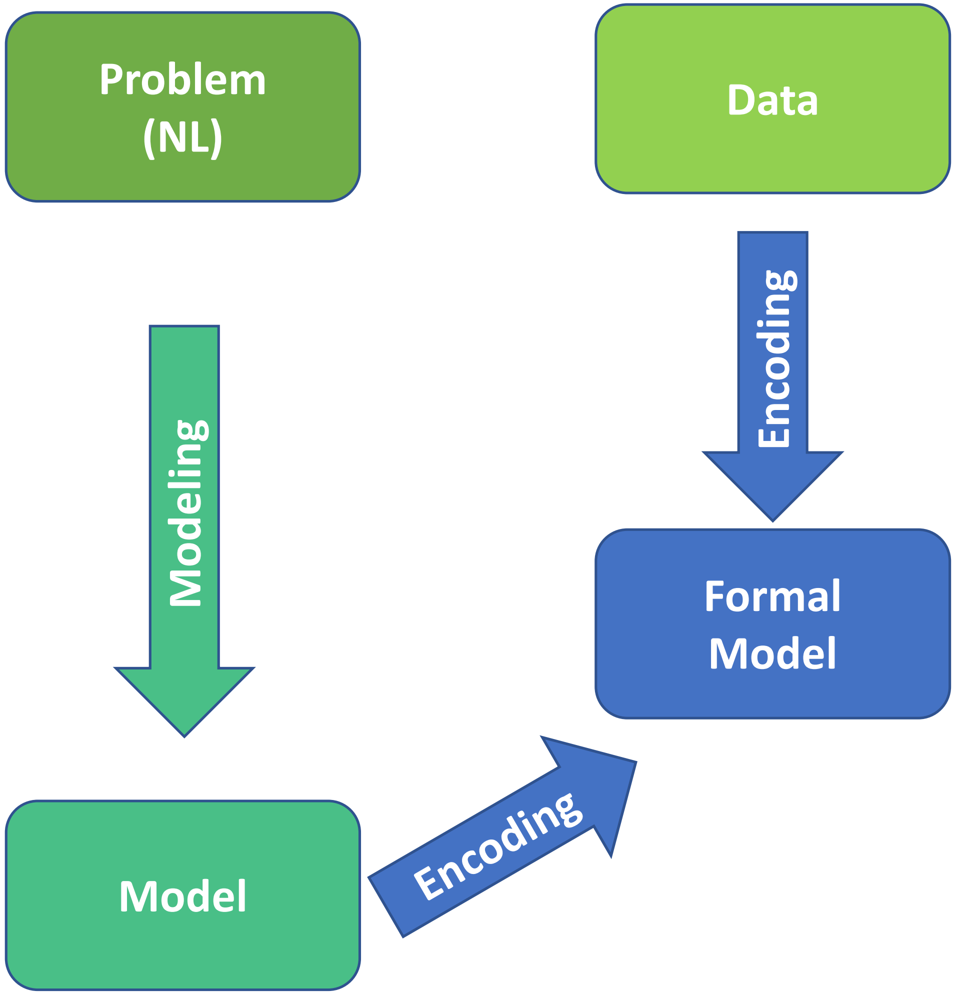
Workflow



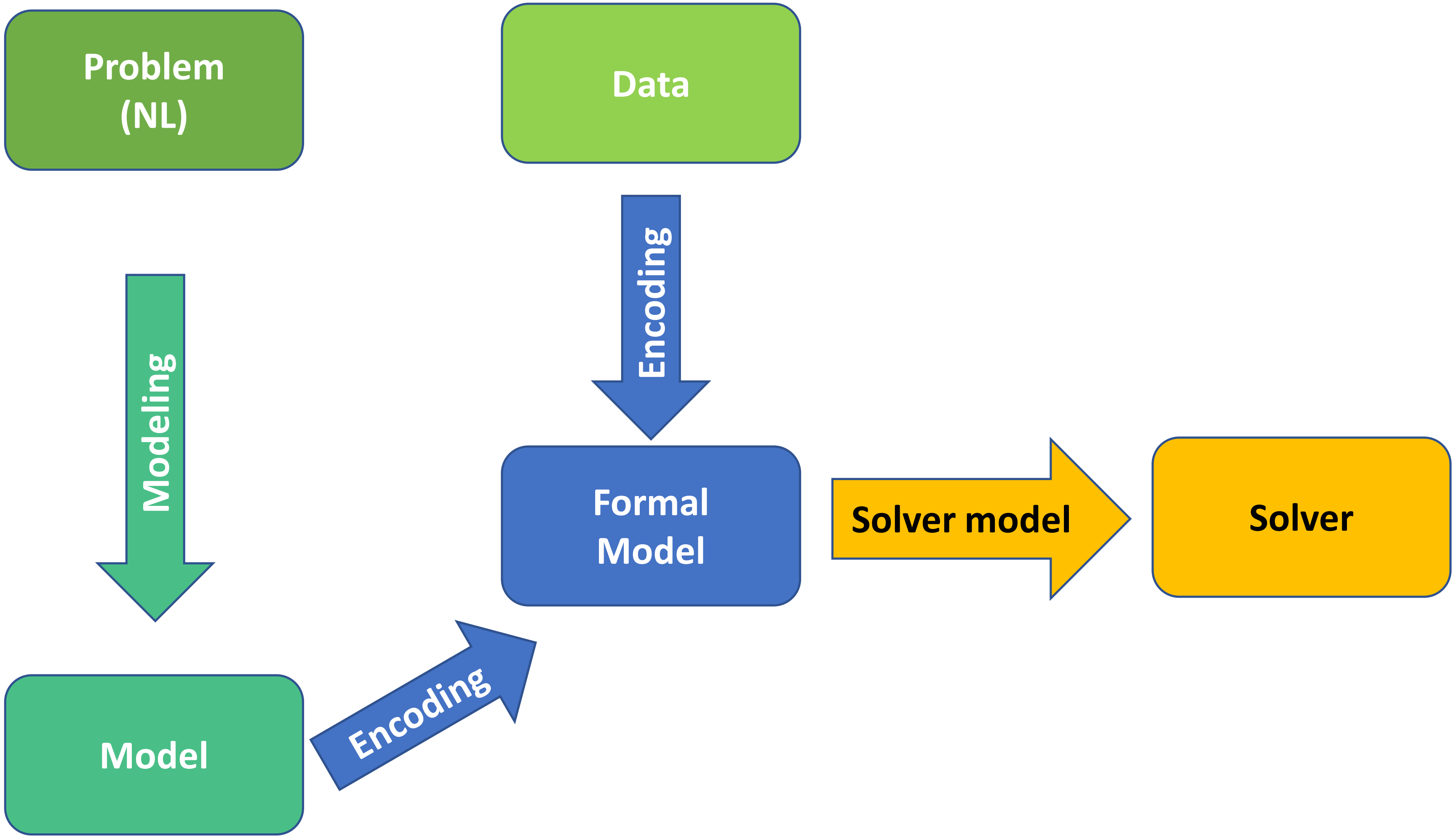
Workflow



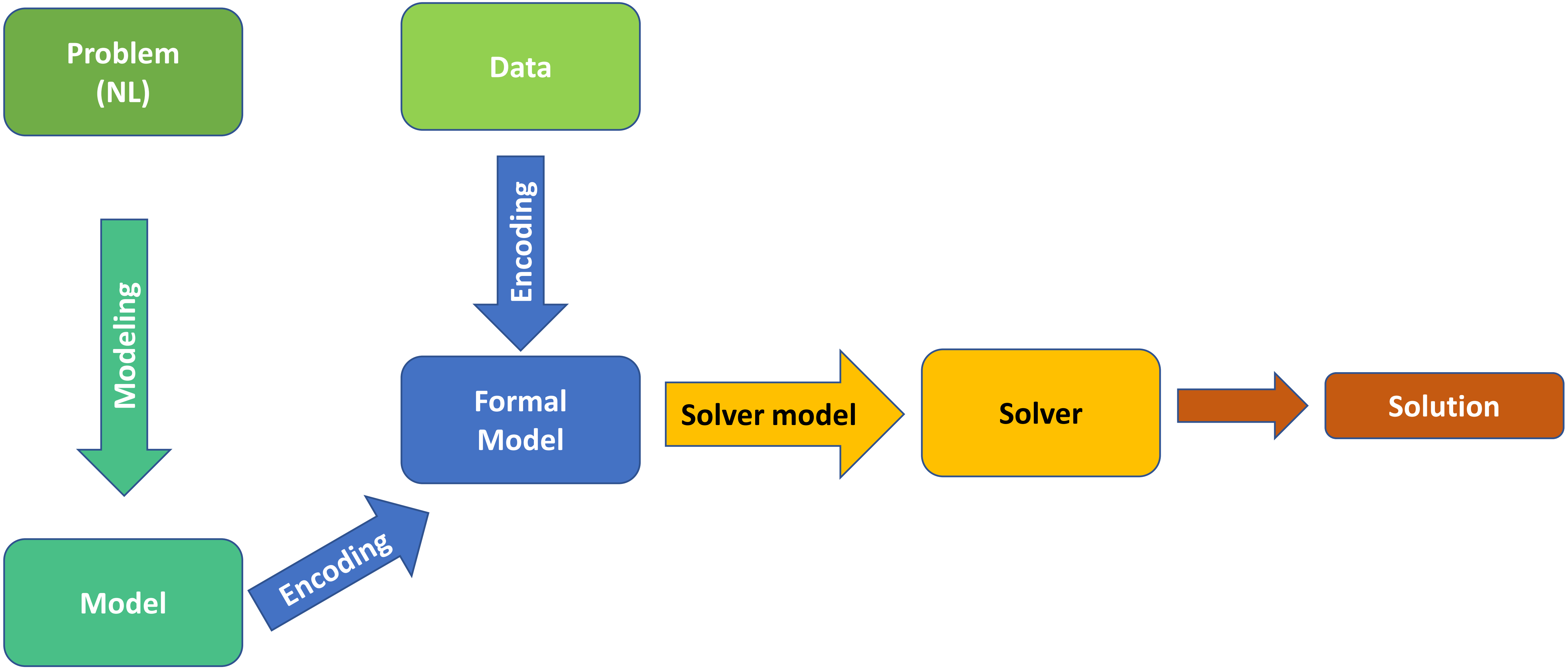
Workflow


Problem
(NL)
Data
Modeling
Encoding
Formal
Model
Model
Encoding

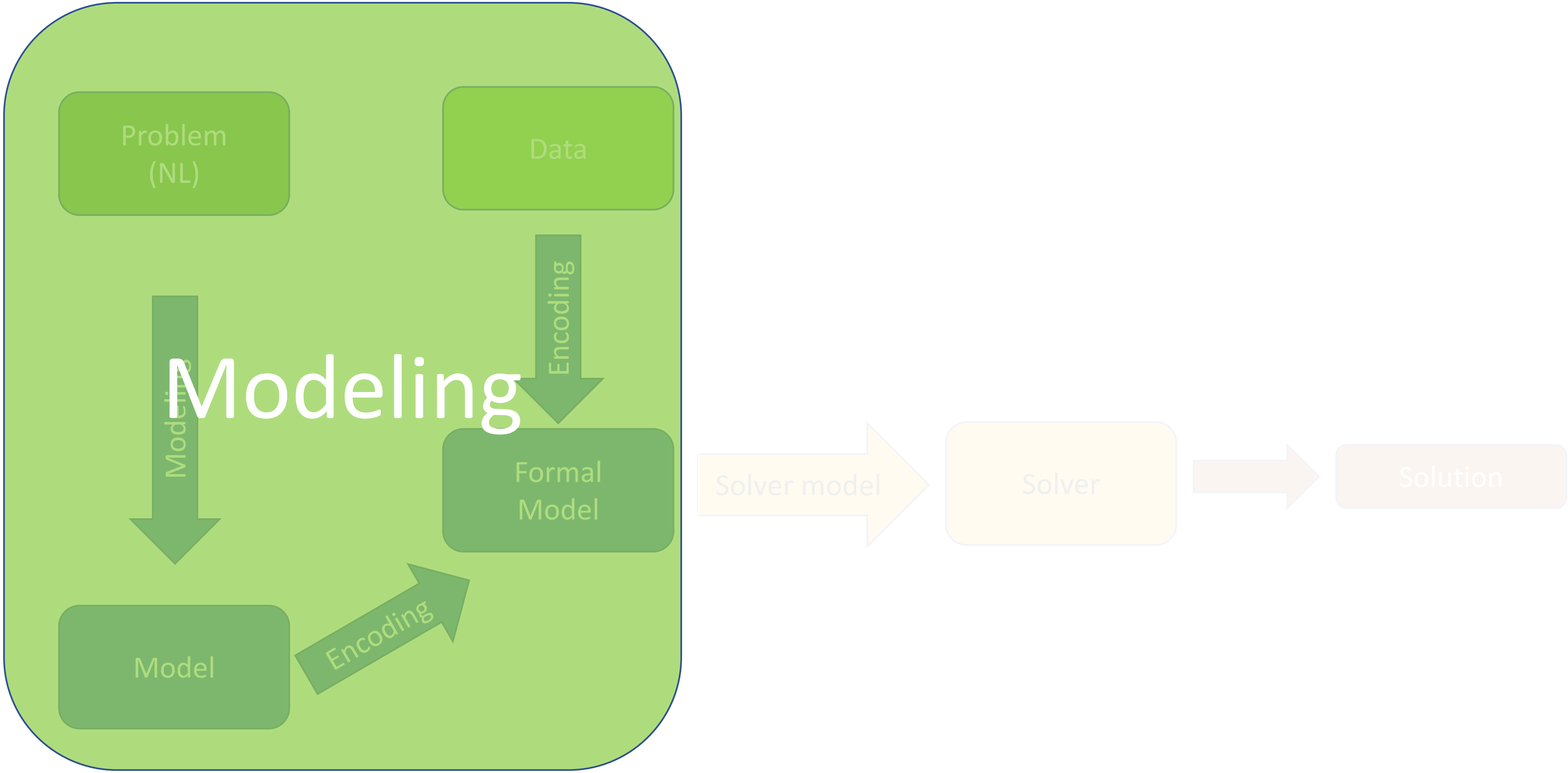
Workflow


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

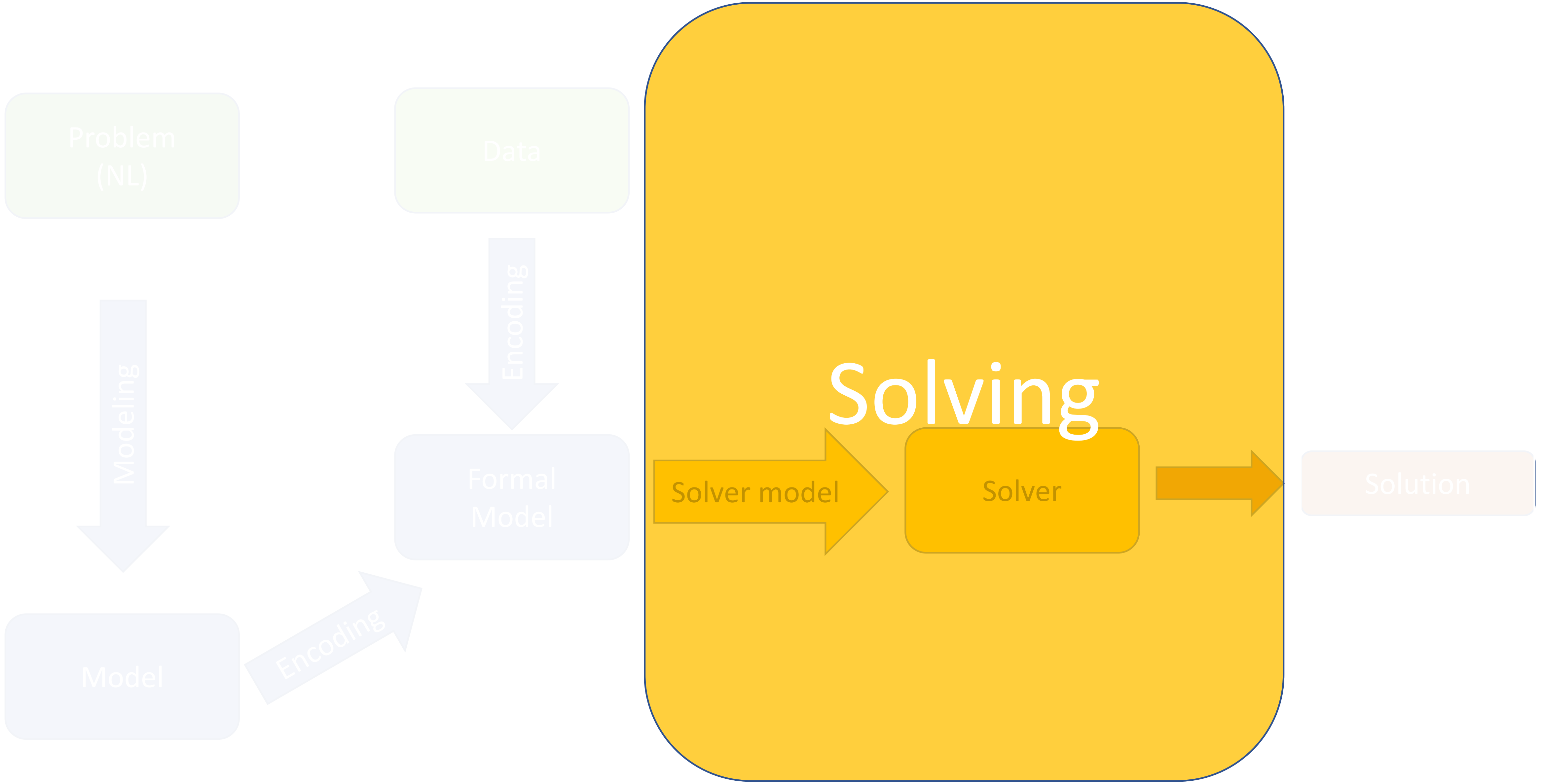
Workflow


Problem
(NL)
Data
Modeling
Encoding
Model
Encoding
Formal
Model
Solver model
Solver
Solution

Workflow

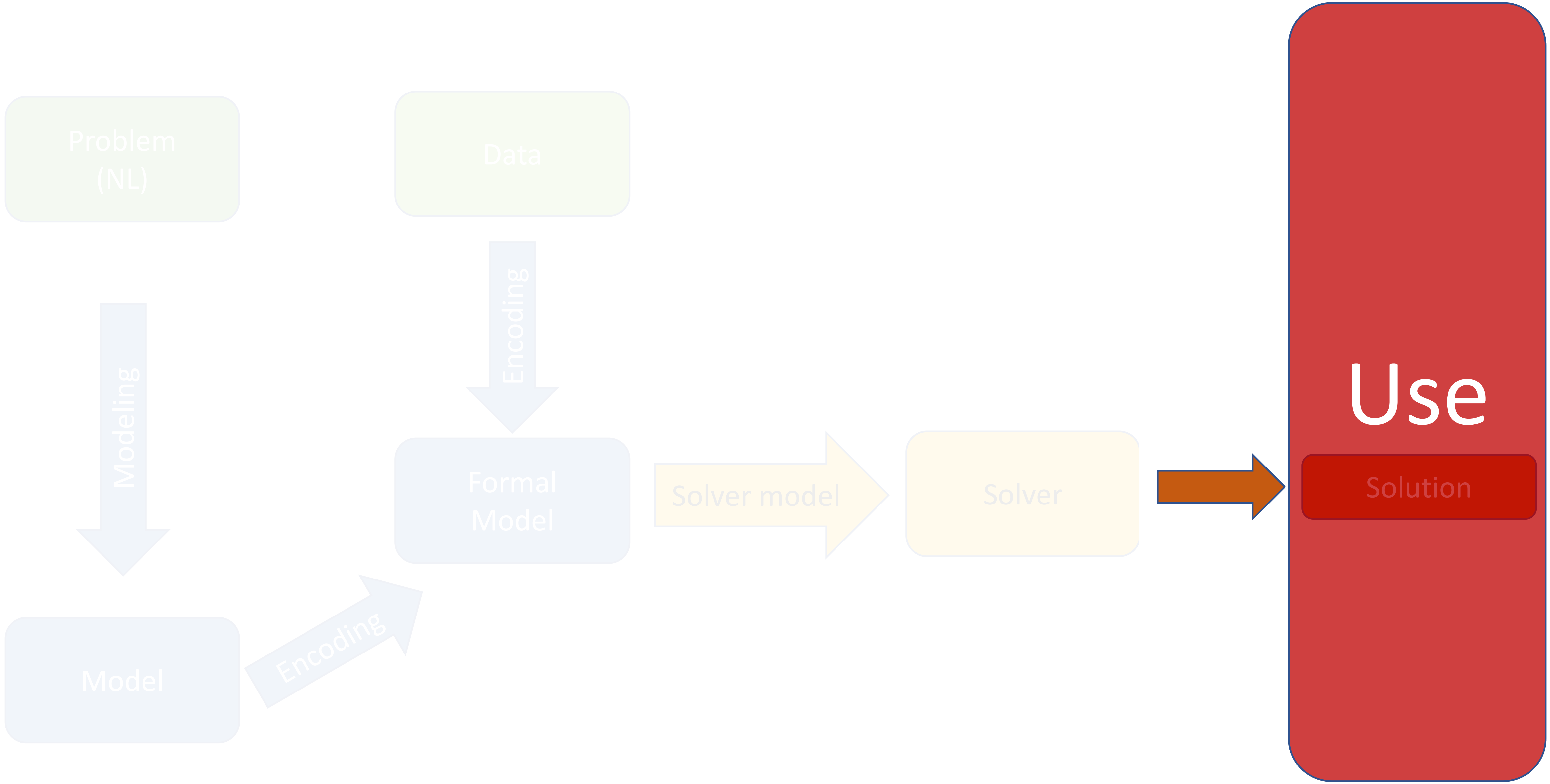

Problem (NL)
Data
Modeling
Encoding
Modeling
Formal Model
Model
Encoding
Solver model
Solver
Solution

Workflow



Problem (NL)
Data
Modeling
Encoding
Formal Model
Model
Encoding
Solving
Solver model
Solver
Solution

Overview



Problem (NL)
Data
Modeling
Encoding
Formal Model
Model
Encoding
Solver model
Solver
Use
Solution

Overview

Modeling

Solving

Use

Bandwidth Allocation Problem (running example)

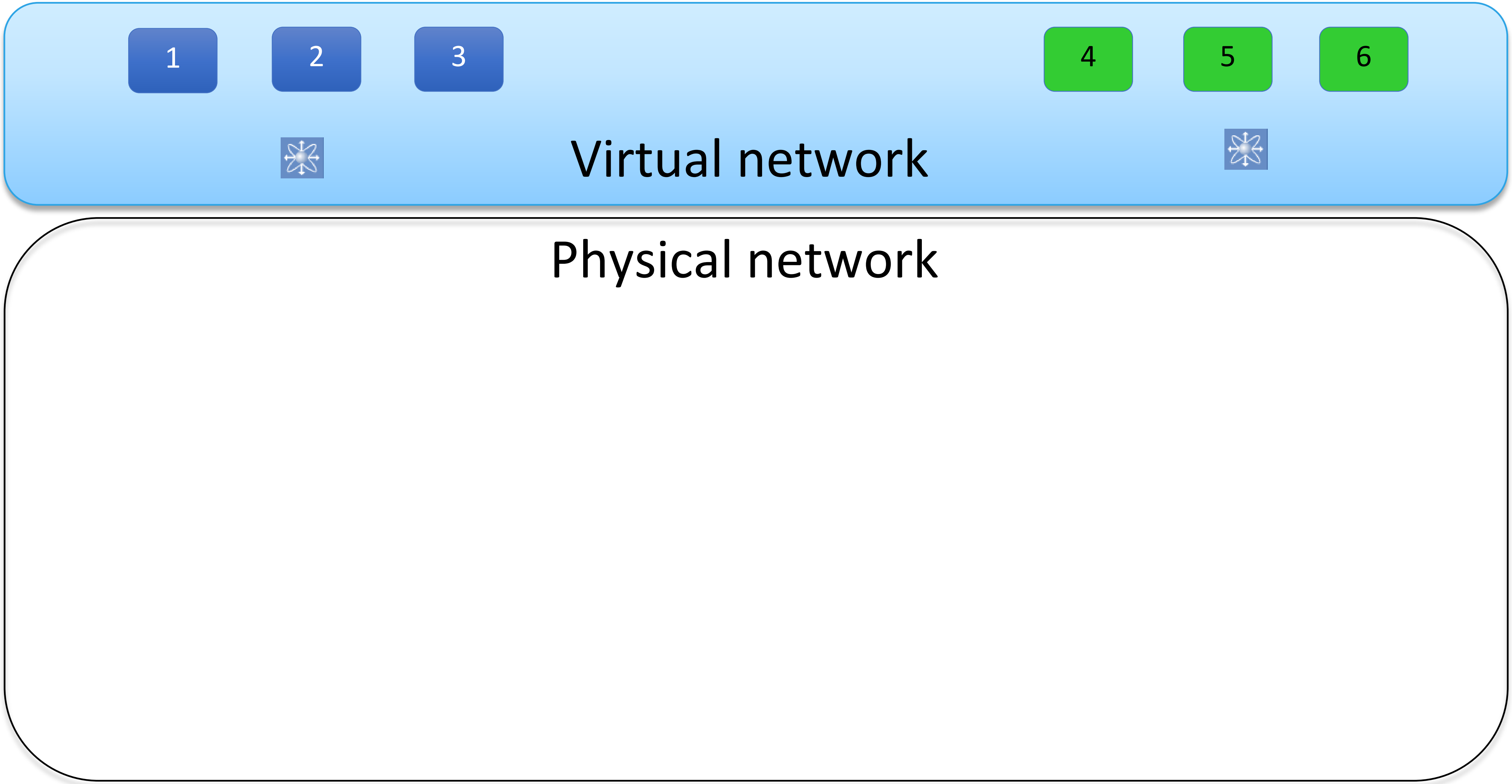
On the feasibility of automation for bandwidth allocation problems in data centers Y. Yuan, A. Wang, R. Alur, B. Thau Loo, FMCAD'2013

Bandwidth Allocation Problem

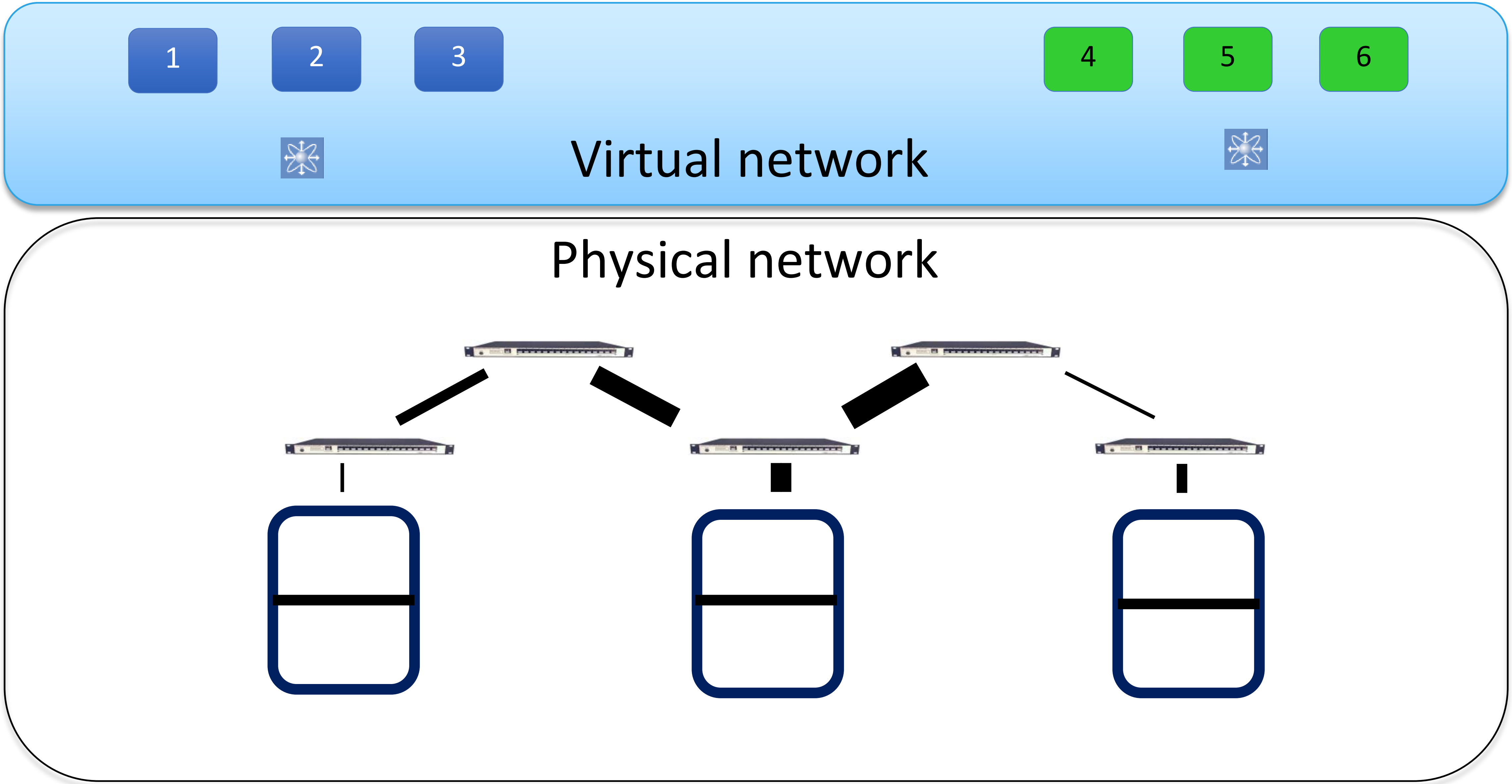
Virtual network

Physical network

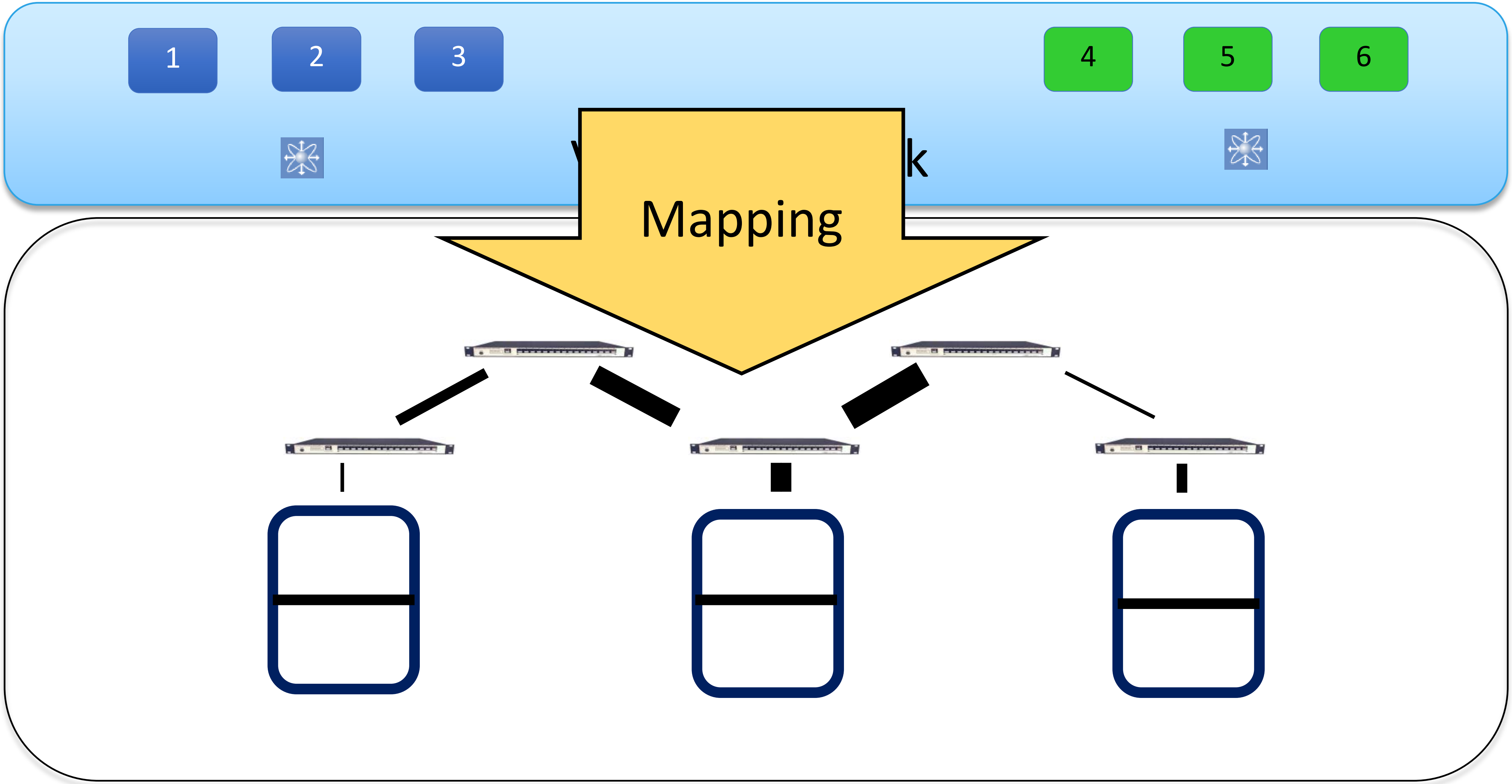
Bandwidth Allocation Problem



Bandwidth Allocation Problem

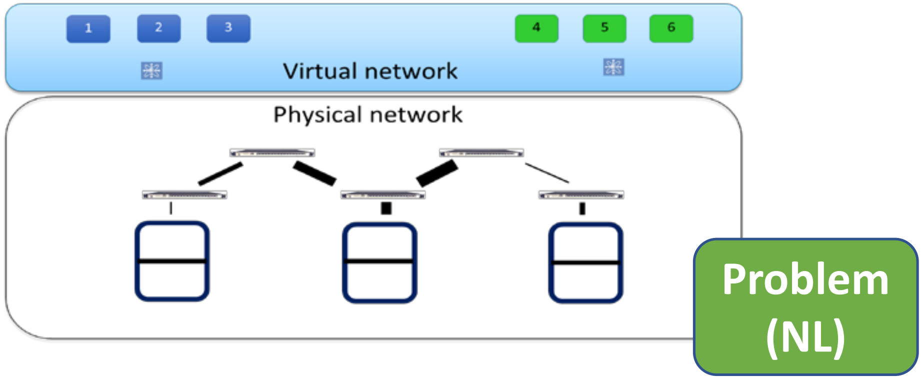


Bandwidth Allocation Problem

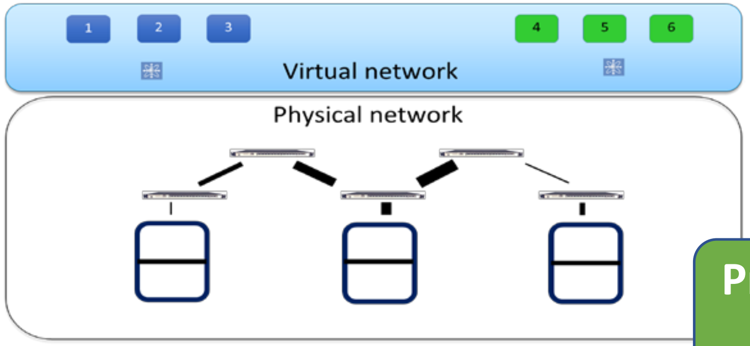


Modeling

Modeling



Modeling

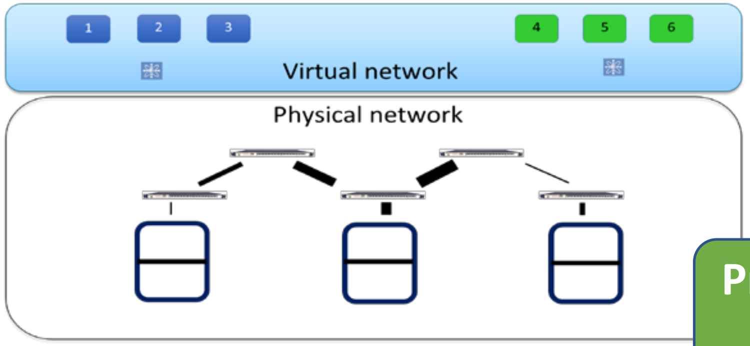


Problem (NL)

Logical constraints:

Model

Modeling



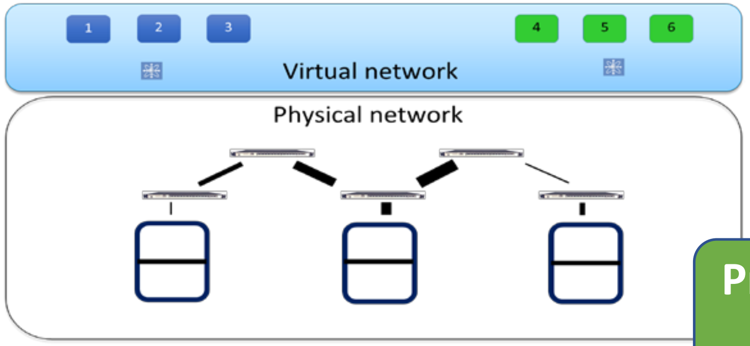
Problem (NL)

Logical constraints:

- each VM is mapped to a host server

Model

Modeling



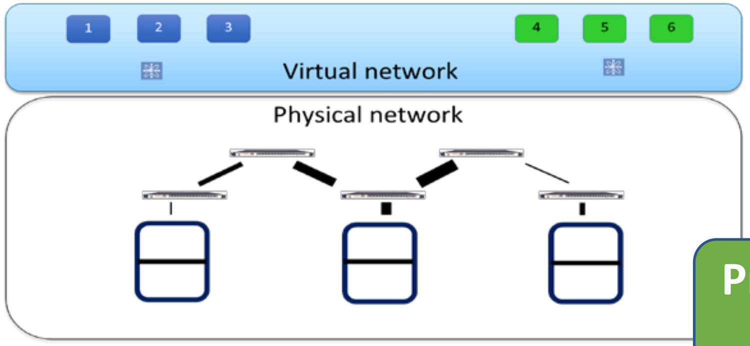
Problem (NL)

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

Model

Modeling



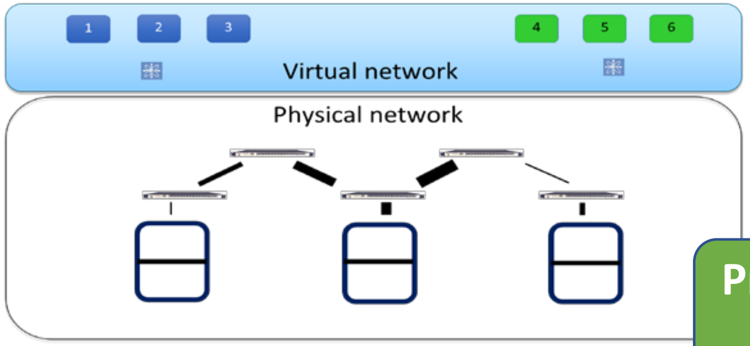
Problem (NL)

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

Model

Modeling



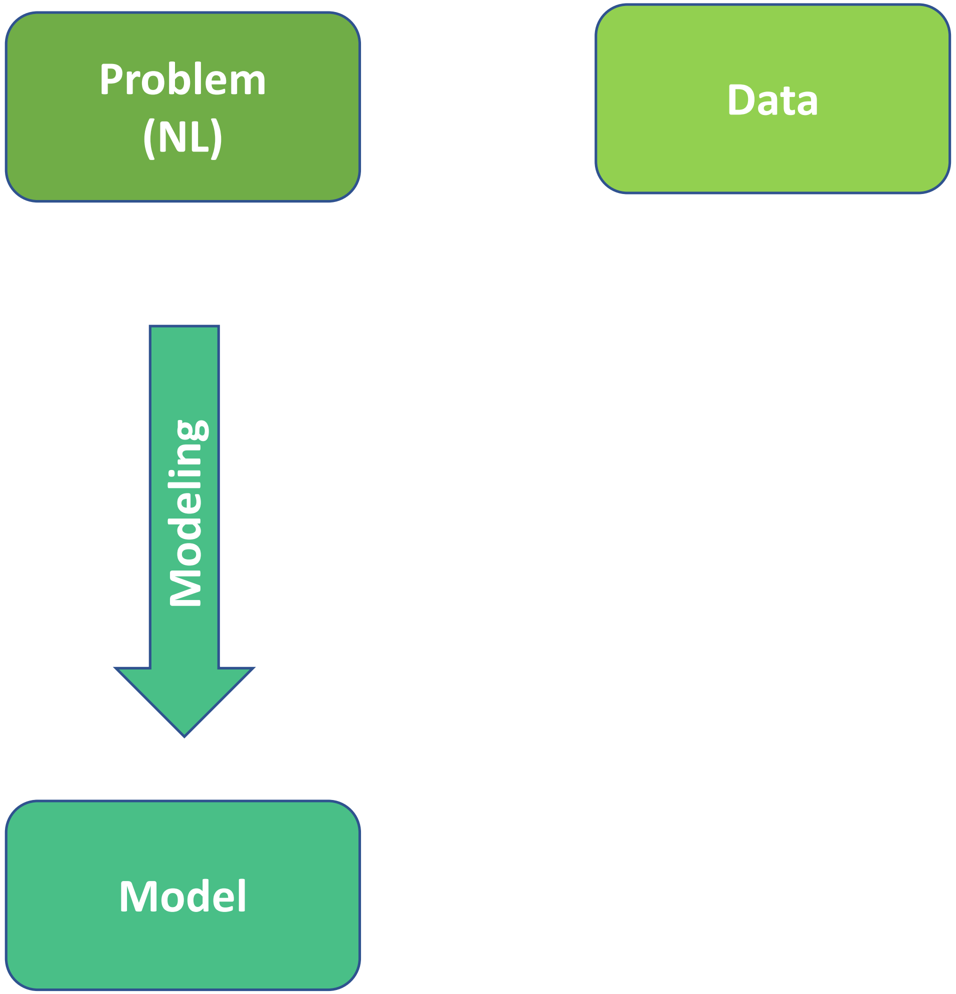
Problem (NL)

Logical constraints:

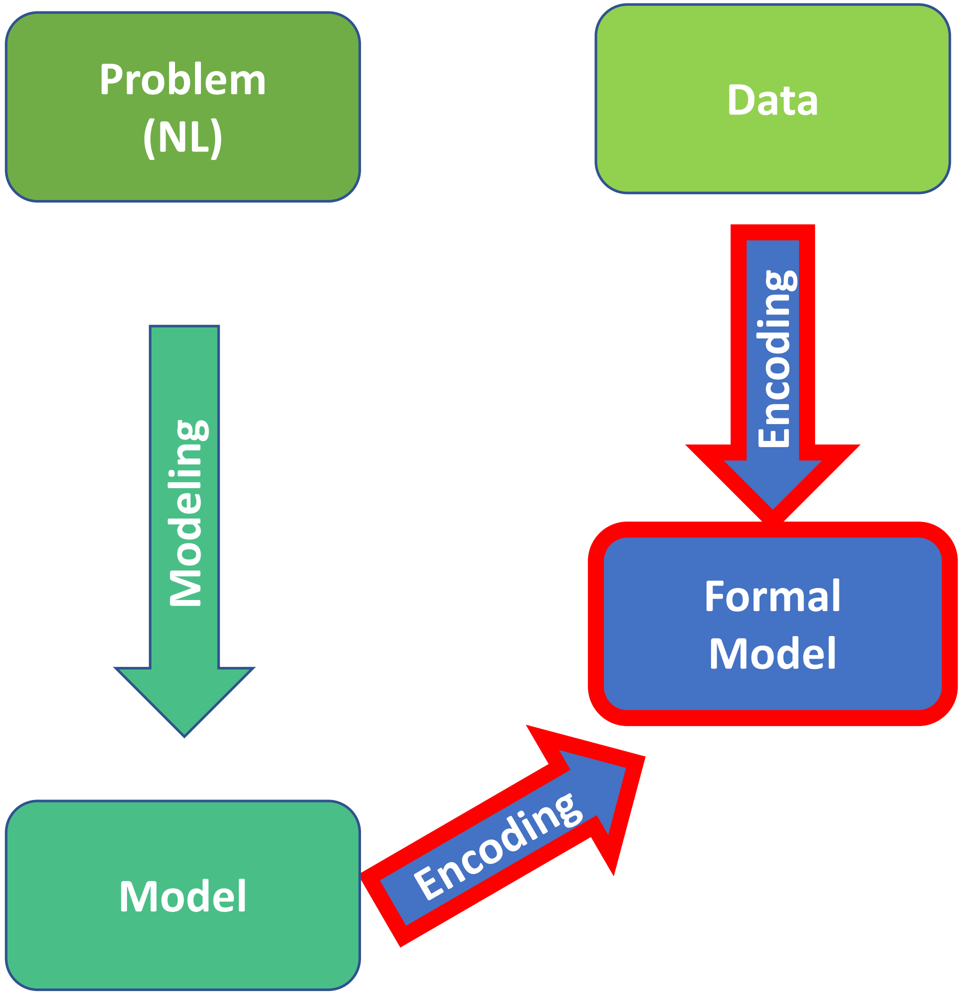
- each VM is mapped to a host server
- for each link between VMs, there is a routing path between the corresponding host servers
- capacity constraints on servers
- capacity constraints on links

Model

Workflow



Workflow


Problem (NL)
Data
Modeling
Encoding
Formal Model
Model
Encoding

Problem modeling

Solvers modeling language

Problem modeling

Solvers modeling language

Inference

Search

Problem modeling

Solvers modeling language

SAT

(T/F)

$$(x_1 \vee \neg x_2) \wedge$$
$$(x_1 \vee \neg x_3) \wedge$$
$$\ldots$$

Inference

Search

Problem modeling

Solvers modeling language

CSP

SAT

(T/F)

$(x_1 \vee \neg x_2) \wedge$
$(x_1 \vee \neg x_3) \wedge$
$\ldots$

Inference

Search

Problem modeling

Solvers modeling language

CSP

MIP

(Int/Real)

$$(2x_1 + x_2 \geq 1) \wedge$$
$$(5x_1 + 4x_2 \leq 4) \wedge$$
$$\dots$$

SAT

(T/F)

$$(x_1 \vee \neg x_2) \wedge$$
$$(x_1 \vee \neg x_3) \wedge$$
$$\dots$$

Inference

Search

Problem modeling

Solvers modeling language

CSP

SMT

(Int/Real/Theory)

MIP

(Int/Real)

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

SAT

(T/F)

$$(x_1 \vee \neg x_2)\wedge$$
$$(x_1 \vee \neg x_3)\wedge$$
$$\ldots$$

Inference

Search

Problem modeling

Solvers modeling language

Fastest black-box solvers

CSP

SMT

(Int/Real/Theory)

MIP

(Int/Real)

$$(2x_1 + x_2 \geq 1) \wedge$$
$$(5x_1 + 4x_2 \leq 4) \wedge$$
$$\dots$$

SAT

(T/F)

$$(x_1 \vee \neg x_2) \wedge$$
$$(x_1 \vee \neg x_3) \wedge$$
$$\dots$$

Inference

Search

Problem modeling

Solvers modeling language

Fast solvers
(for verification)

CSP

SMT

(Int/Real/Theory)

MIP

(Int/Real)

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

SAT

(T/F)

$$(x_1 \vee \neg x_2) \wedge$$
$$(x_1 \vee \neg x_3) \wedge$$
$$\ldots$$

Inference

Search

Problem modeling

Solvers modeling language

Fast solvers for highly structured problems

CSP

SMT

(Int/Real/Theory)

MIP

(Int/Real)

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

SAT

(T/F)

$$(x_1 \vee \neg x_2)\wedge$$
$$(x_1 \vee \neg x_3)\wedge$$
$$\ldots$$

Inference

Search

Problem modeling

Solvers modeling language

CSP

SMT

(Int/Real/Theory)

MIP

(Int/Real)

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

SAT

(T/F)

$(x_1 \vee \neg x_2) \wedge$
$(x_1 \vee \neg x_3) \wedge$
$\ldots$

Inference

Search

SAT solvers

SAT solvers

Consists of a set of Boolean variables and clauses

x_1, x_2, x_3 T F $\neg x_i$

SAT solvers

Consists of a set of Boolean variables and clauses

x_1, x_2, x_3 T F $\neg x_i$

$C_1 = (x_1)$ $C_3 = (x_1 \vee x_2)$

$C_2 = (x_2)$ $C_4 = (\neg x_1 \vee \neg x_3)$

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

$$C_1 = (x_1) \qquad C_3 = (x_1 \vee x_2)$$

$$C_2 = (x_2) \qquad C_4 = (\neg x_1 \vee \neg x_3)$$

SAT solvers

Consists of a set of Boolean variables and clauses

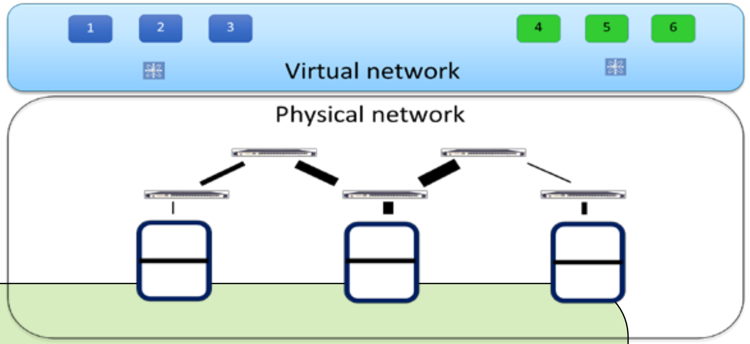
$$x_1, x_2, x_3$$

T T F

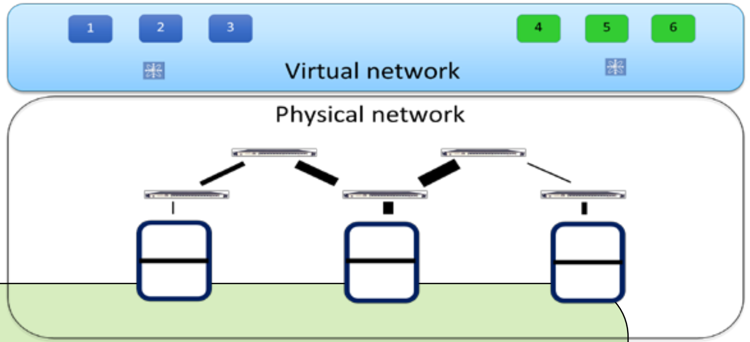
$$C_1 = (x_1) \qquad C_3 = (x_1 \vee x_2)$$

$$C_2 = (x_2) \qquad C_4 = (\neg x_1 \vee \neg x_3)$$

Bandwidth Allocation Problem



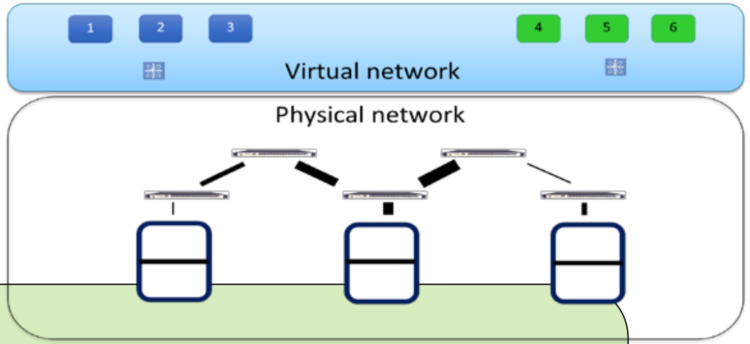
Bandwidth Allocation Problem



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

$$X(v,s) = 1 \text{ iff } v \text{ is hosted in } s$$

Bandwidth Allocation Problem



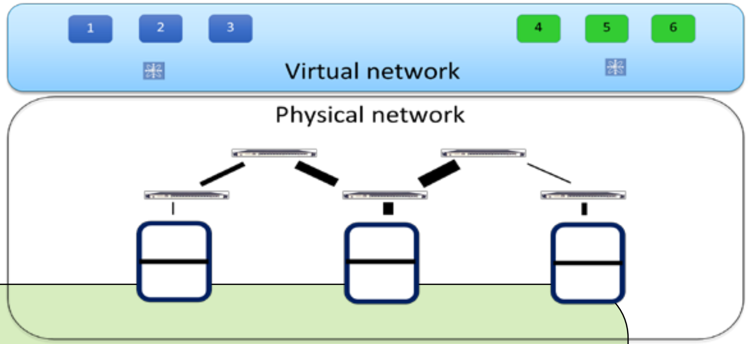
$$\forall v \in \text{VM}, \forall s \in \text{Server } X(v,s) \in \{0,1\}$$

$$X(v,s) = 1 \text{ iff } v \text{ is hosted in } s$$

(1) each VM is mapped to a host server

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

Bandwidth Allocation Problem



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

$$X(v,s) = 1 \text{ iff } v \text{ is hosted in } s$$

(1) each VM is mapped to a host server

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

(3) capacity constraints on servers

$$\bigwedge_{s \in \text{Servers}} \left(\sum_{v \in \text{ VM}} X(v,s) \leq \text{capacity}(s)\right)$$

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

Solvers modeling language

CSP

SMT

(Int/Real/Theory)

MIP

(Int/Real)

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

SAT

(T/F)

$(x_1 \vee \neg x_2) \wedge$
$(x_1 \vee \neg x_3) \wedge$
$\ldots$

Inference

Search

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

$$x_1, x_2, x_3$$

$$AllDifferent(x_1, x_2, \ldots, x_n)$$

$$Regular(x_1, x_2, \ldots, x_n, A)$$

CSP solvers

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

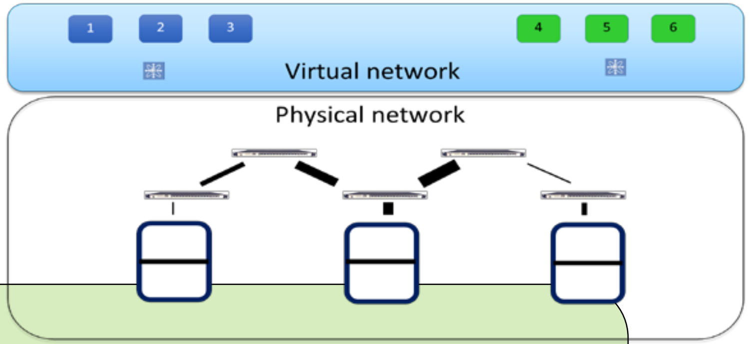
$$x_1, x_2, x_3$$

Goal: find an assignment that satisfies all constraints

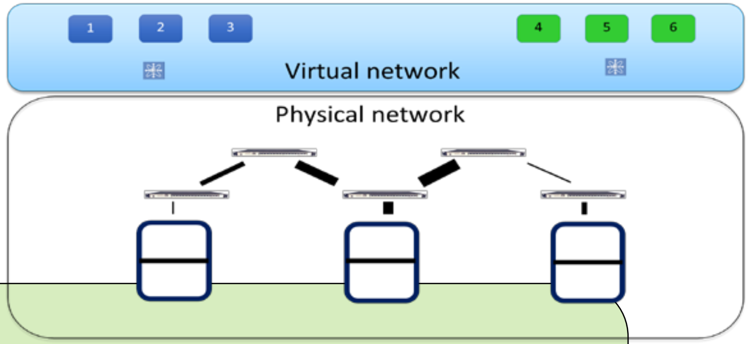
$$AllDifferent(x_1, x_2, \ldots, x_n)$$

$$Regular(x_1, x_2, \ldots, x_n, A)$$

Bandwidth Allocation Problem



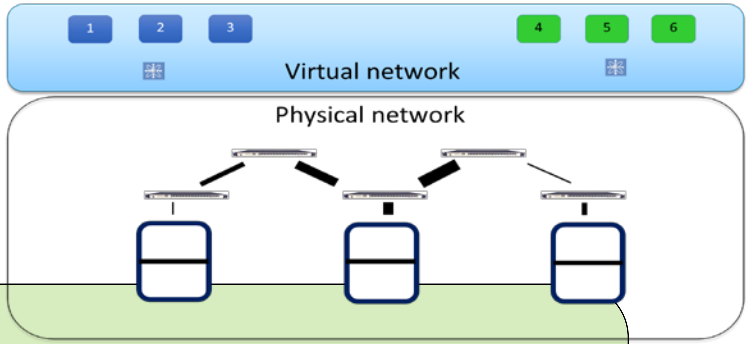
Bandwidth Allocation Problem



$$\forall v \in \mathrm{VM}, X(v) \in \{1, 2, 3\}$$

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

Bandwidth Allocation Problem



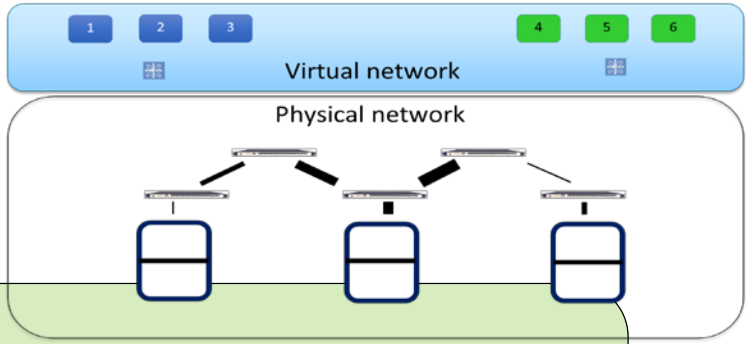
$$\forall v \in \text{VM}, X(v) \in \{1, 2, 3\}$$

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

(1) each VM is mapped to a host server

No need for a constraint

Bandwidth Allocation Problem



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

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

(1) each VM is mapped to a host server

No need for a constraint

(3) capacity constraints on servers

$$GlobalCardConstraint[(X_1, \ldots, X_n), [\text{capacity}(s_1), \ldots, \text{capacity}(s_m)]]$$

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

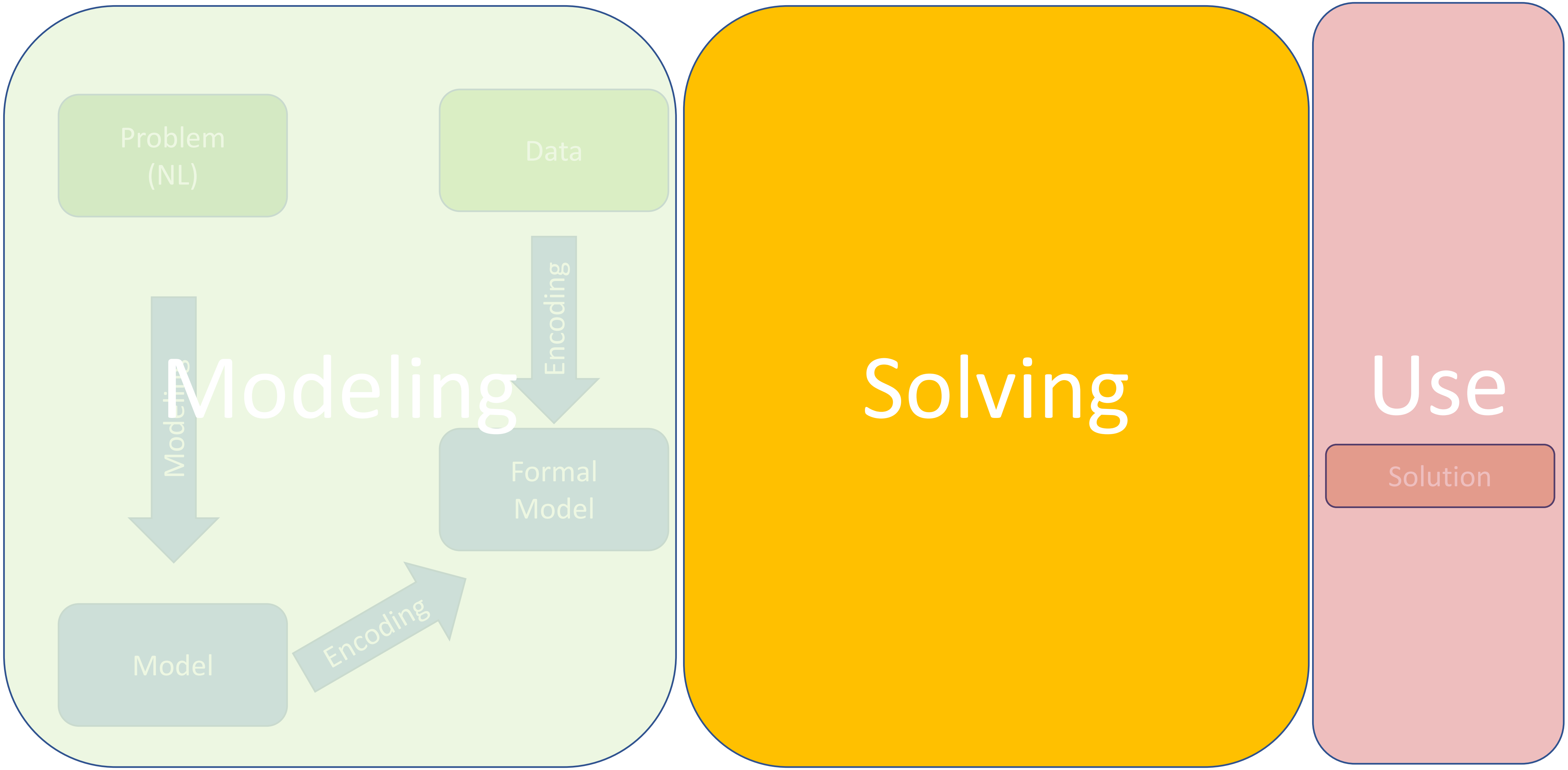
Which solver to use?

- 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

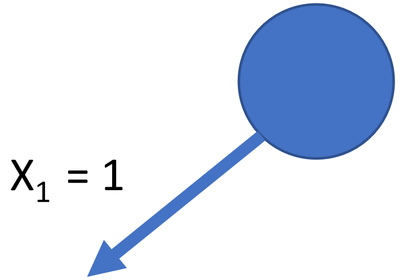
Which solver to use?

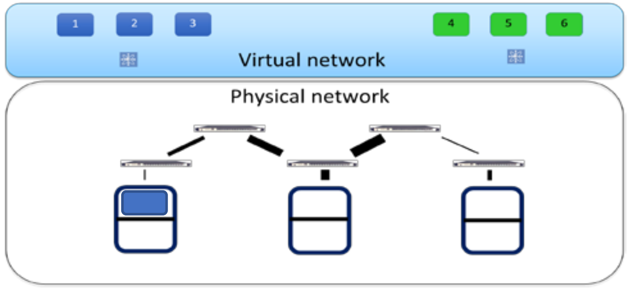
- Start with CP model. Use the simplest model possible.
  Most likely it will be slow.
- Take advantage of the domain specific information
  Remove model symmetry, problem decomposition, heuristics
- Avoid using complicated variables, e.g. set variables
  It is very hard to reason about them efficiently
- Relax constraints (e.g. use soft constraints instead of hard constraints)

Overview

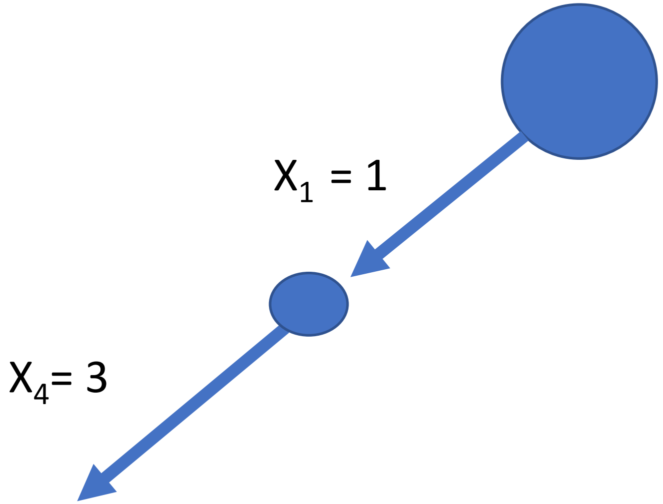

Problem (NL)
Data
Modeling
Encoding
Modeling
Formal Model
Model
Encoding
Solving
Use
Solution

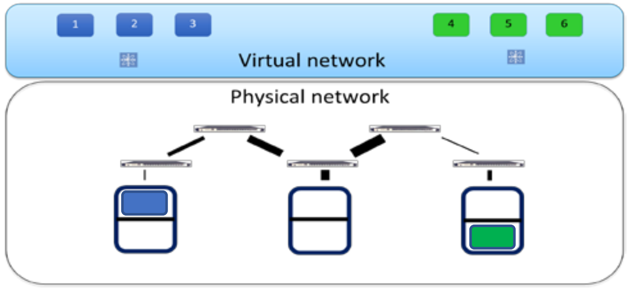
Backtracking search



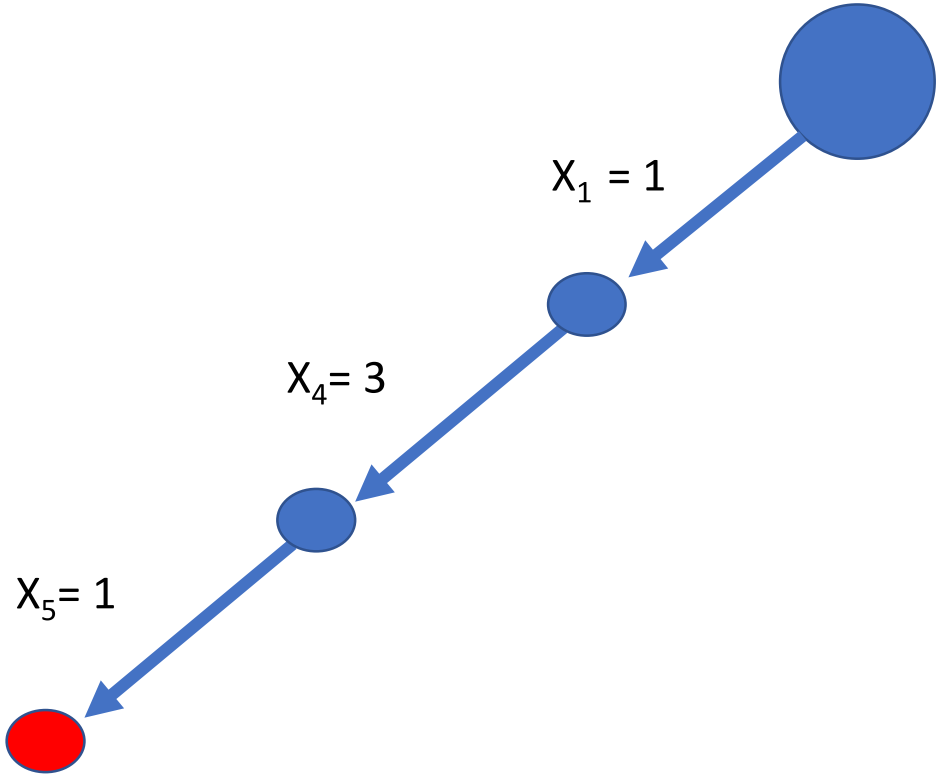


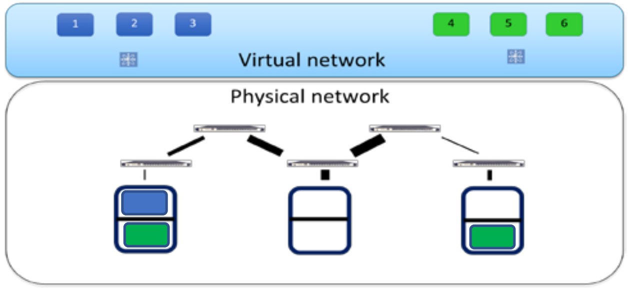
Backtracking search



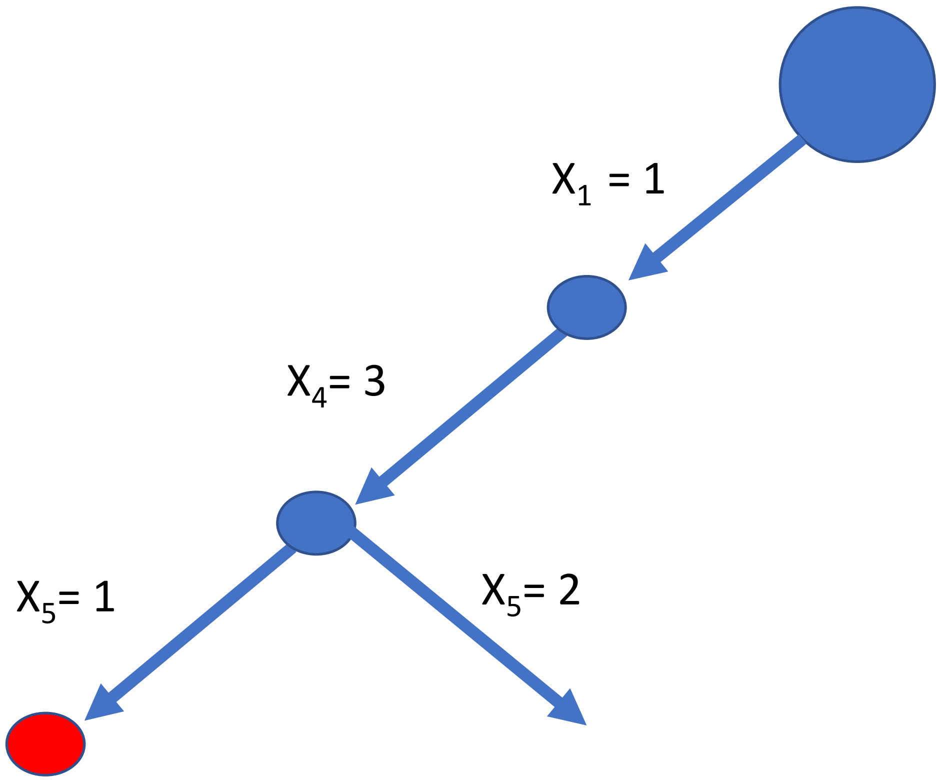


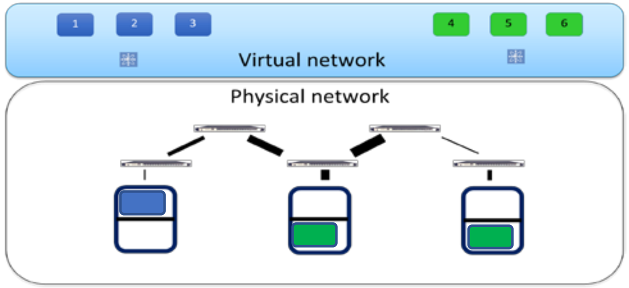
Backtracking search



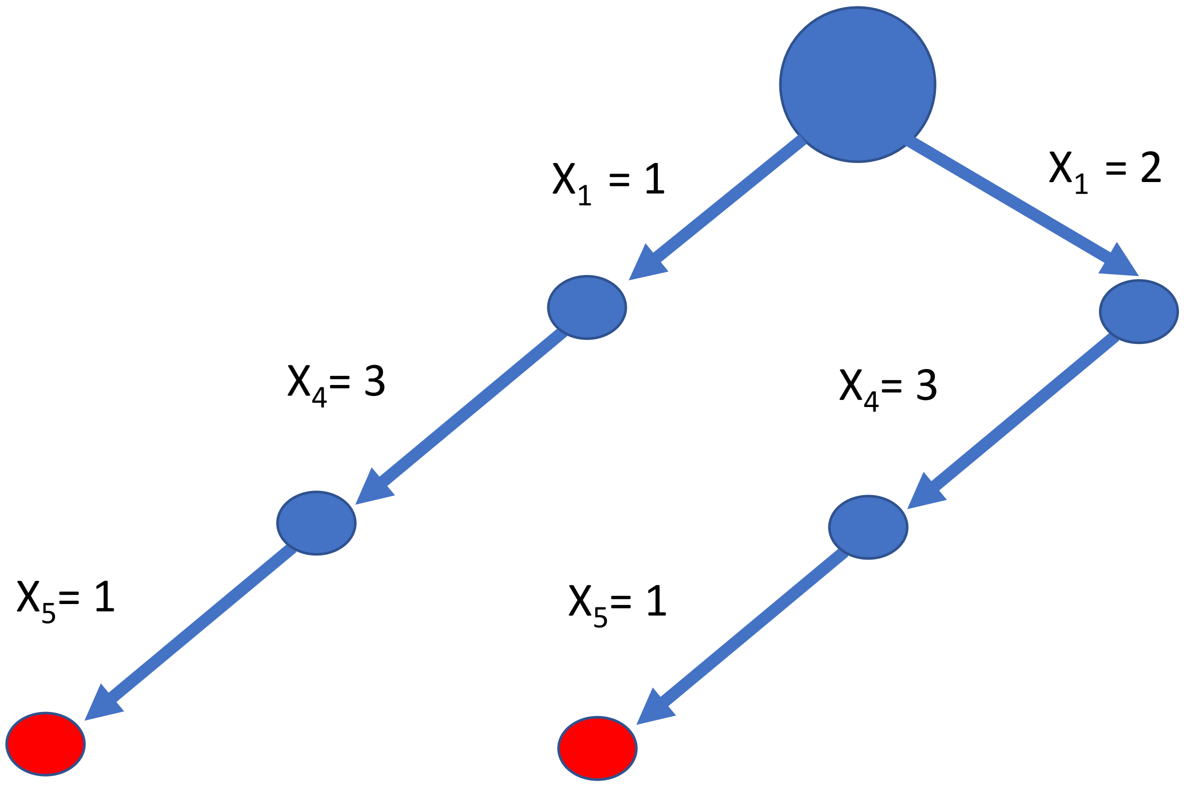


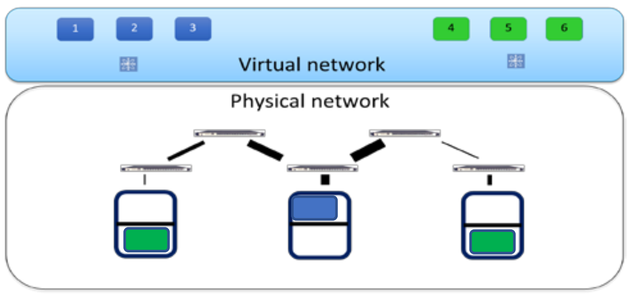
Backtracking search



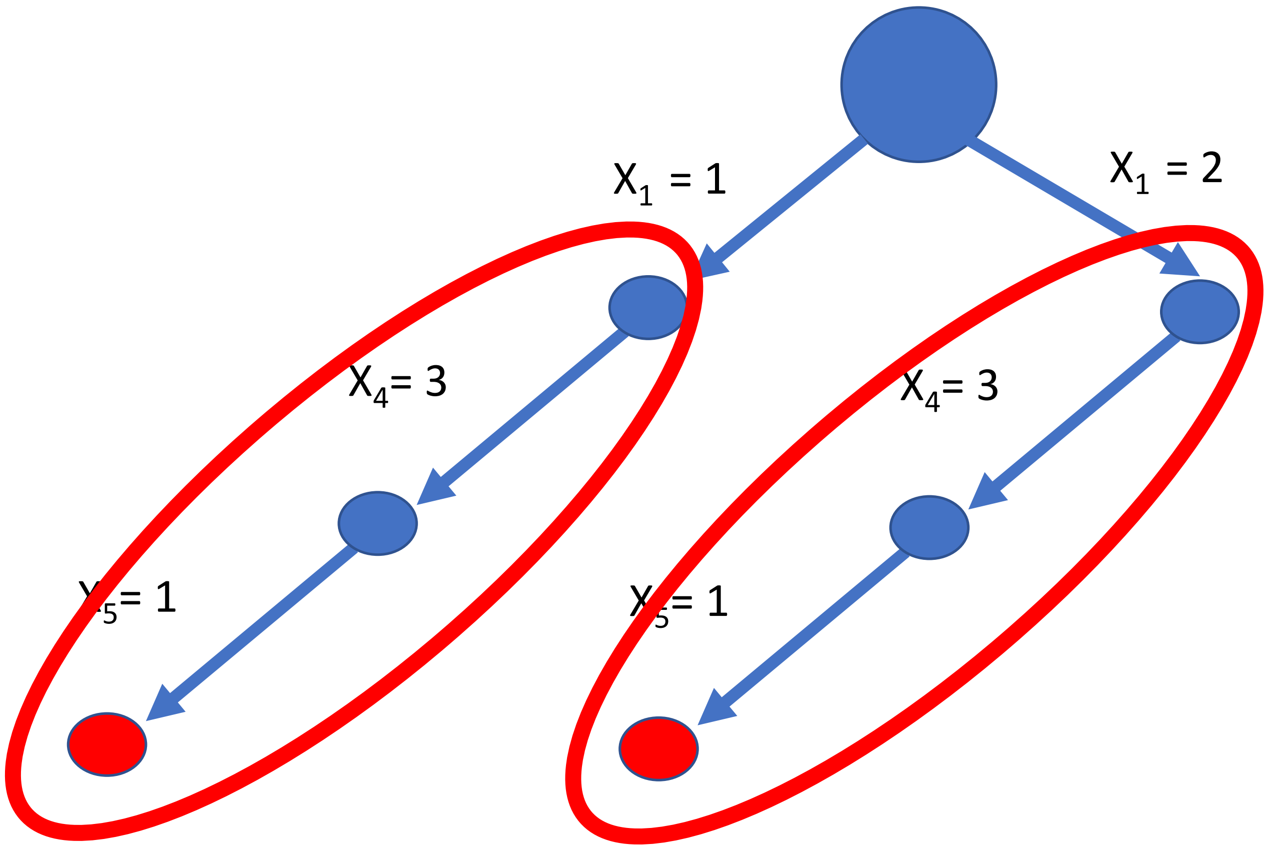


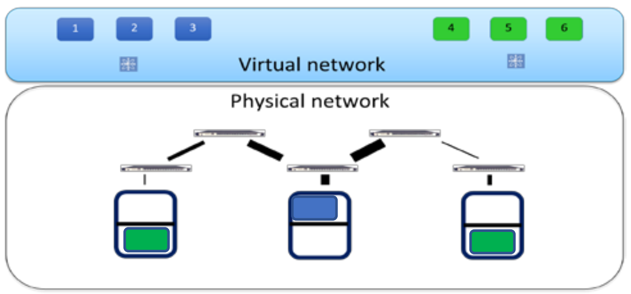
Backtracking search



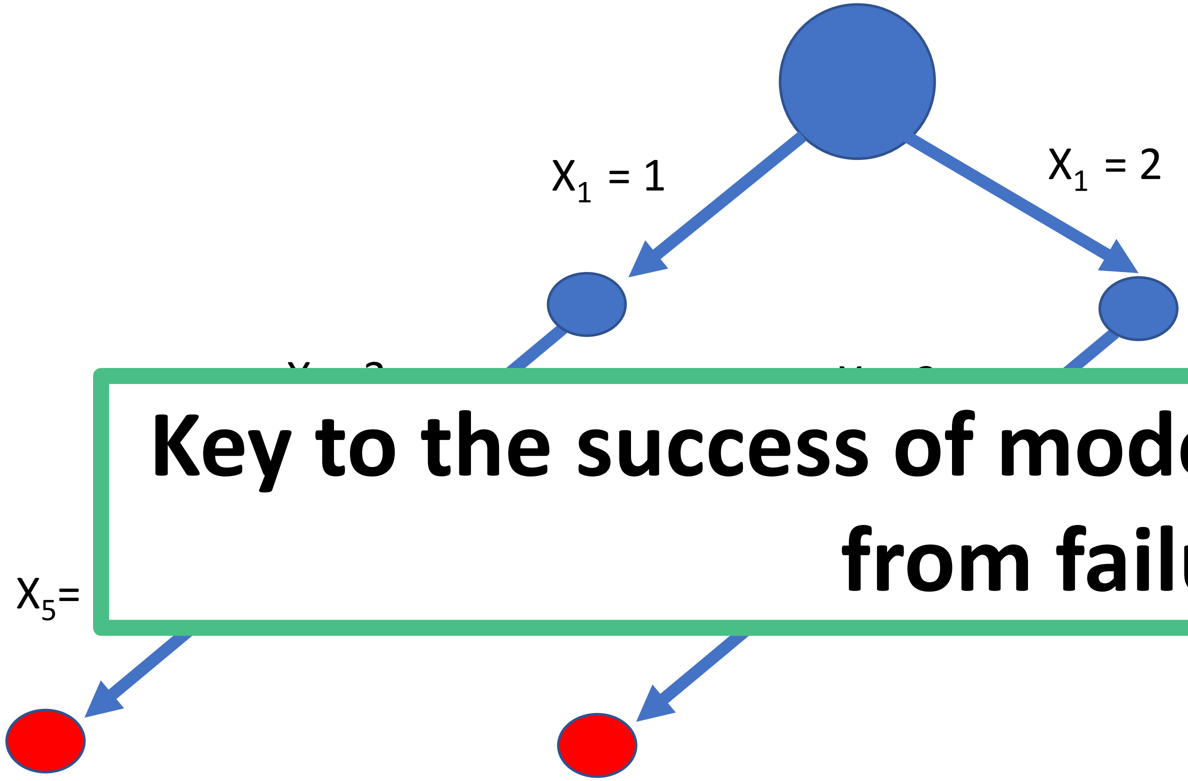


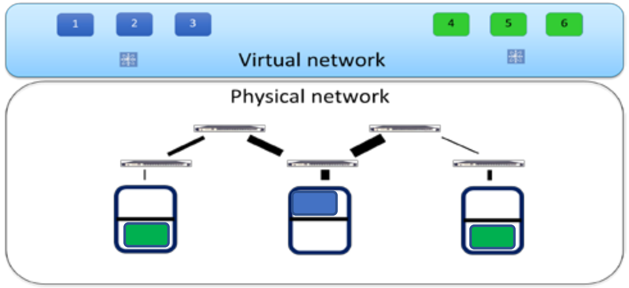
Backtracking search





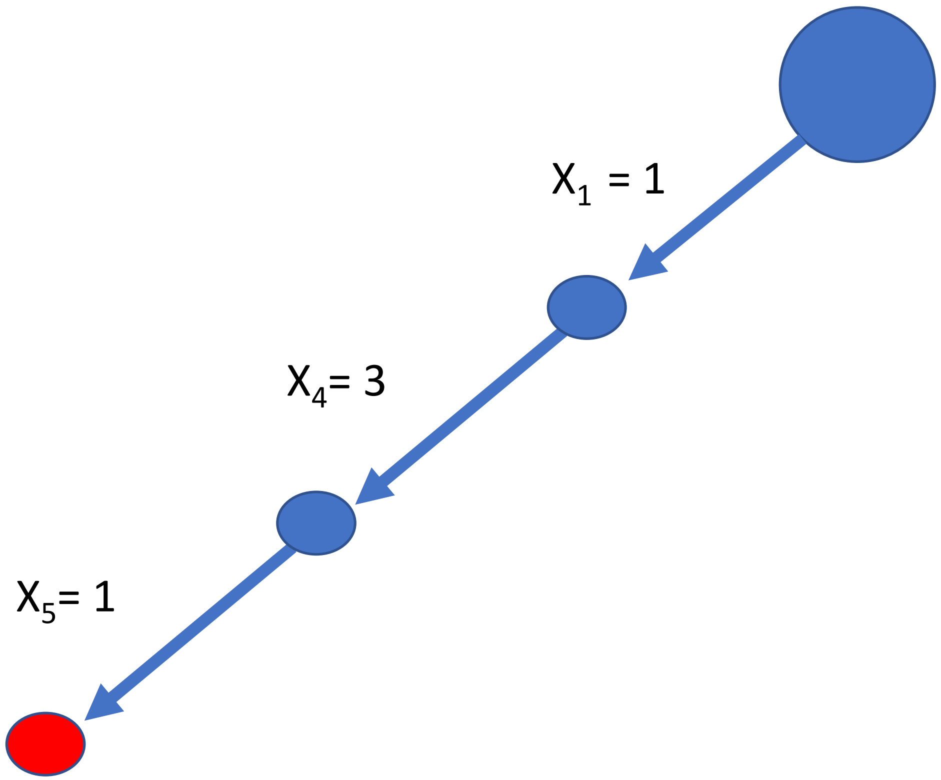
Backtracking search

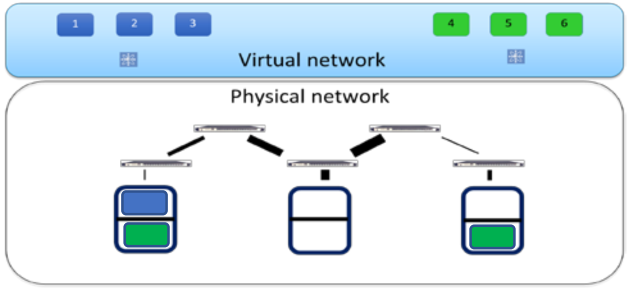




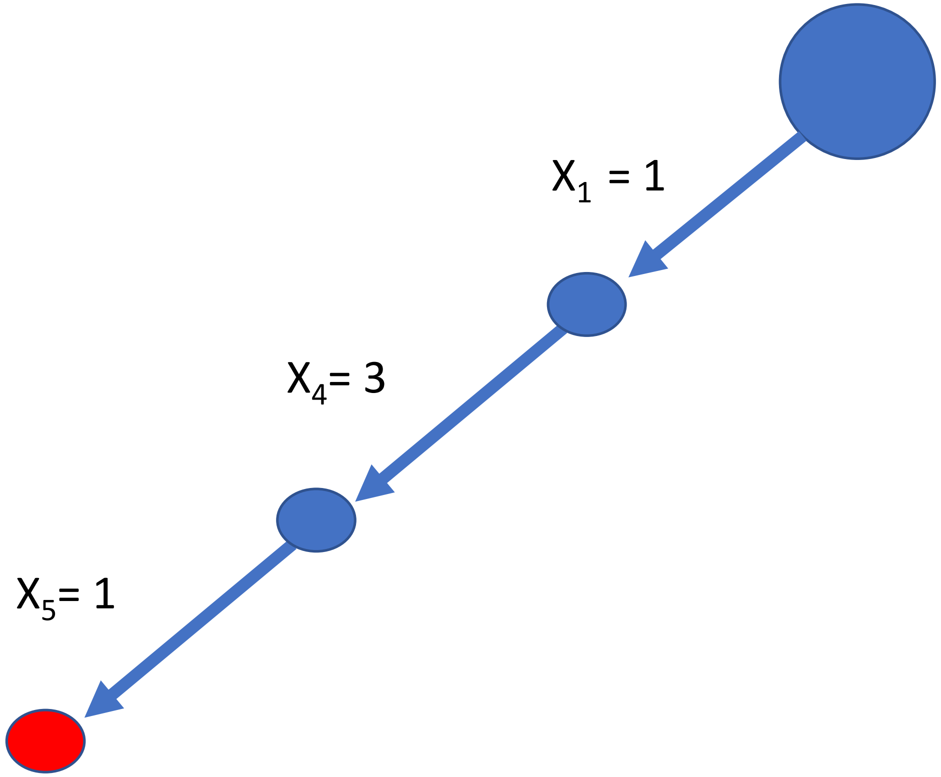
Key to the success of modern solvers is learning from failures

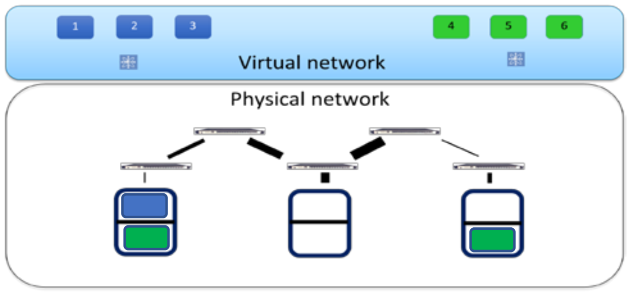
Learning mechanism





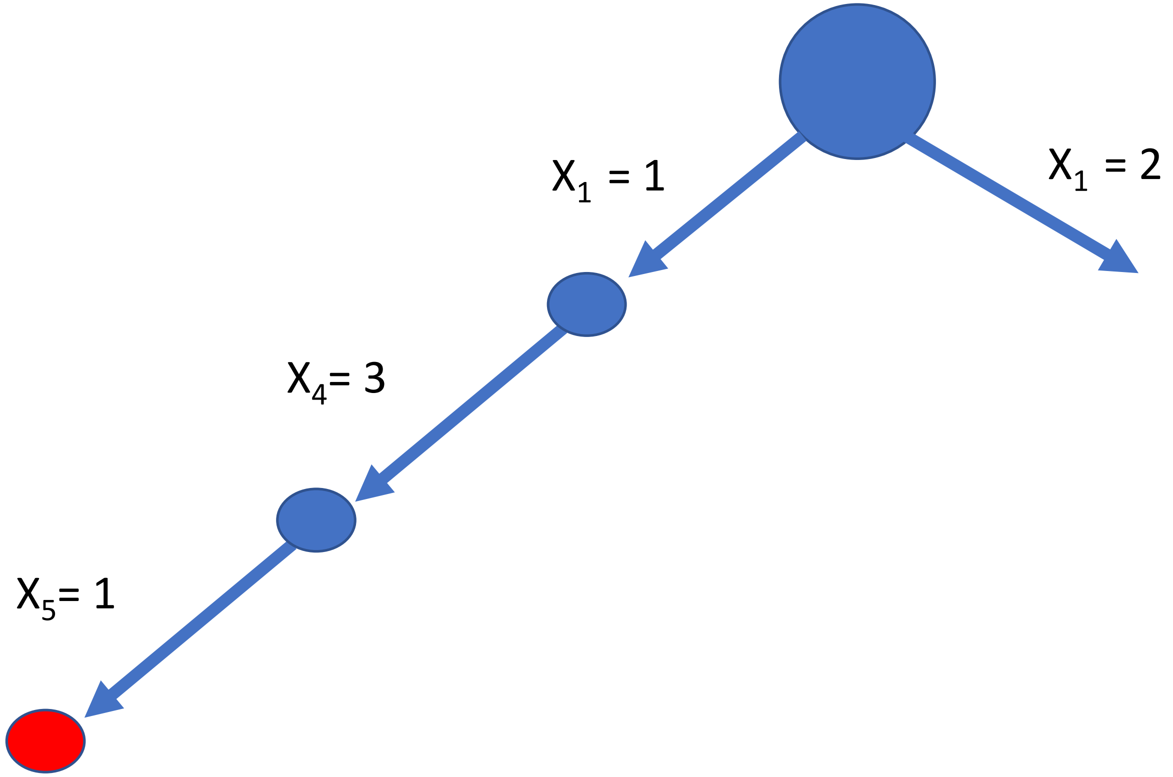
Learning mechanism

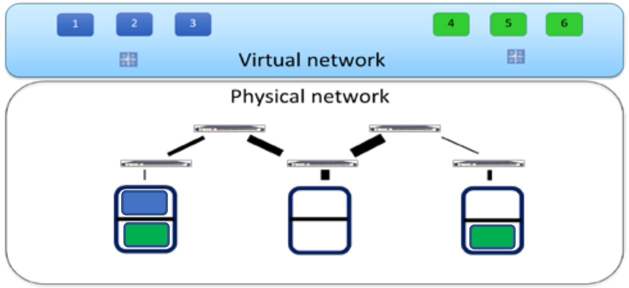




NOT ($X_4 = 3$ AND $X_5 = 1$)

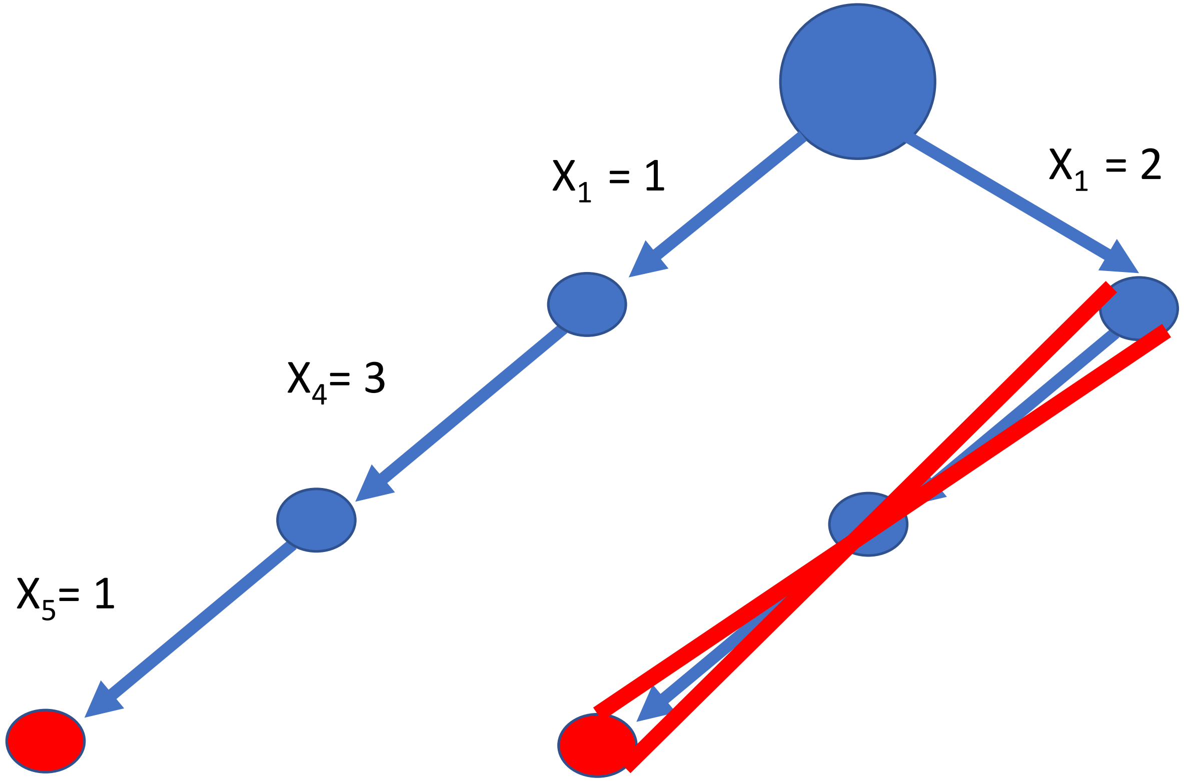
Learning mechanism

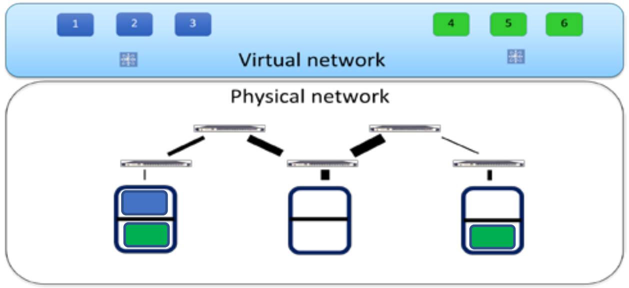




NOT ($X_4 = 3$ AND $X_5 = 1$)

Learning mechanism





NOT (X_4= 3 AND X_5= 1)

CP solvers best learning model

High level model
(CP)

CP solvers best learning model

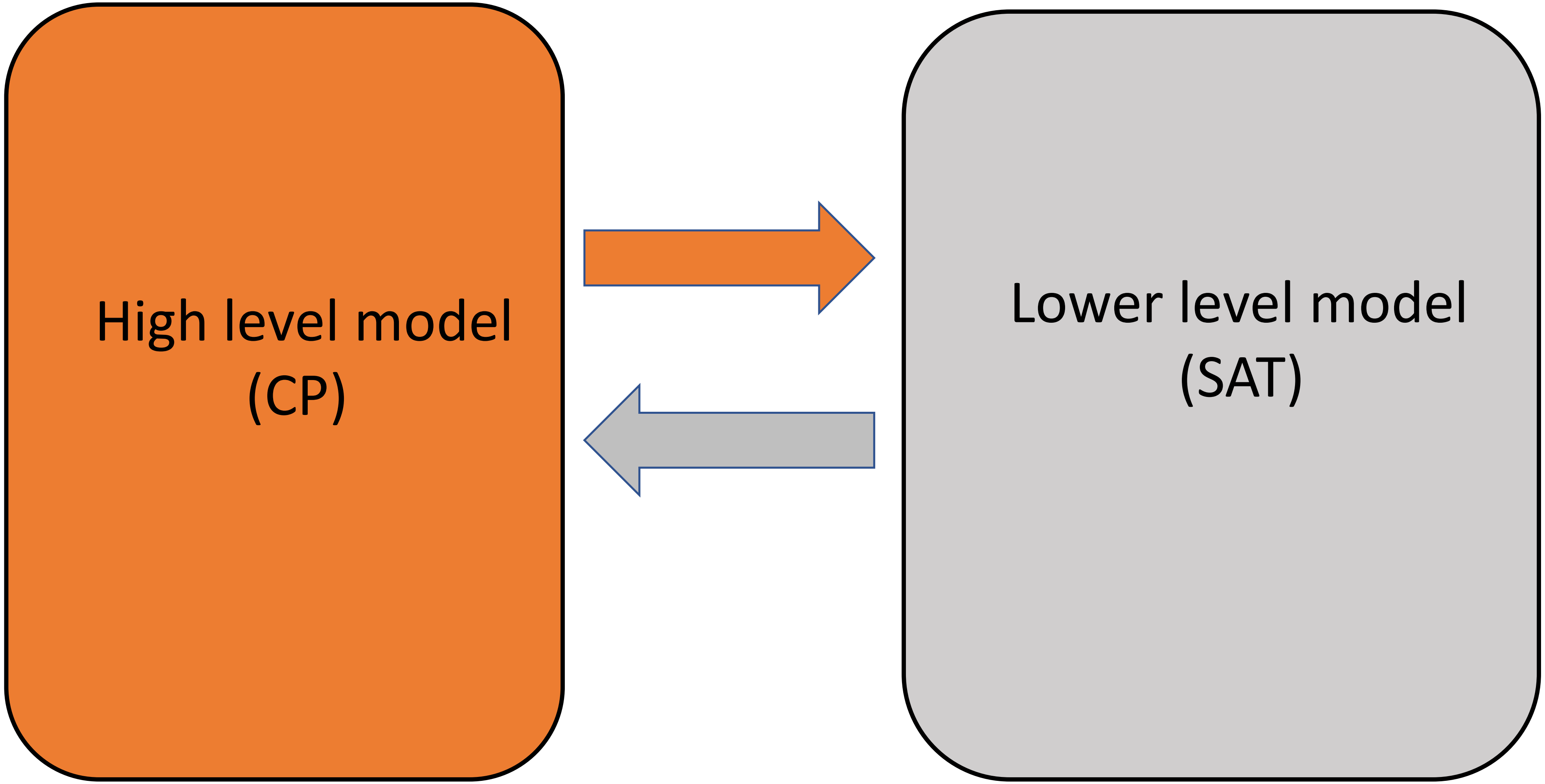
High level model
(CP)

Lower level model
(SAT)

CP solvers best learning model

High level model (CP) $\neq$ Lower level model (SAT)

CP solvers best learning model



CP solvers best learning model

AllDifferent(X,Y,Z)

X , Y ϵ {1,2}
Z ϵ {1,2,3}

SAT

CP solvers best learning model

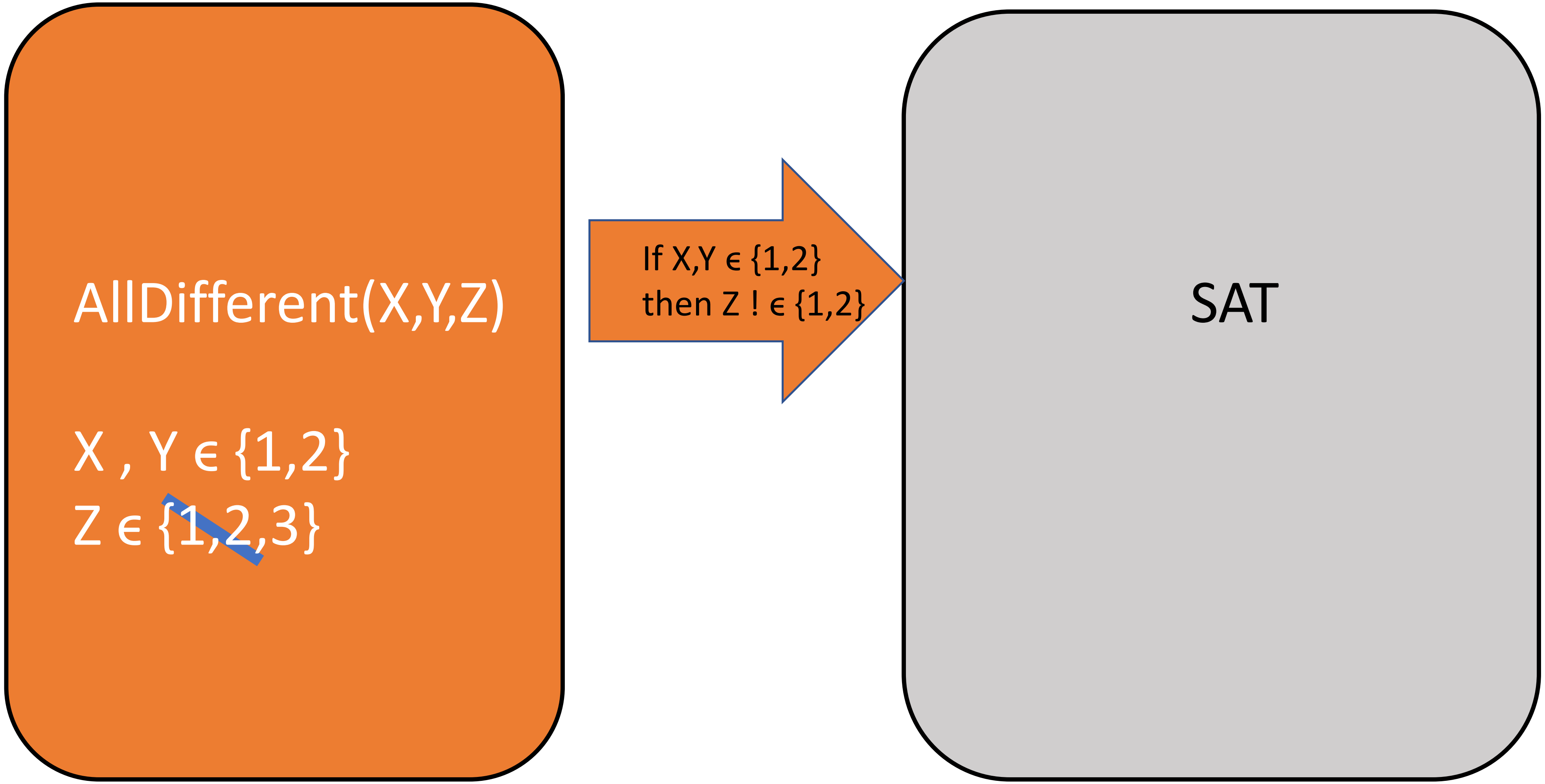
AllDifferent(X,Y,Z)

X , Y ∈ {1,2}

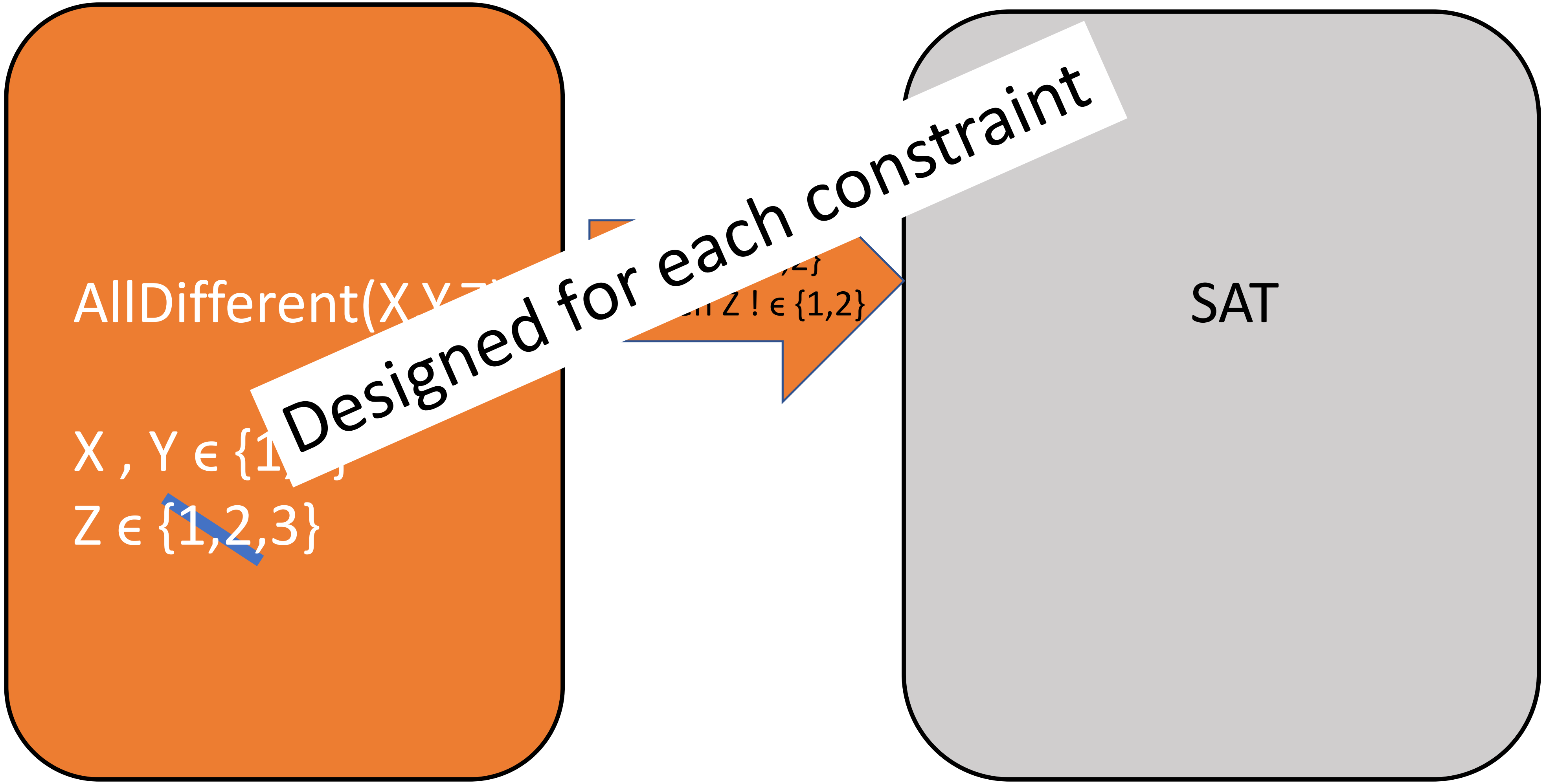
Z ∈ {~~1,2,~~3}

SAT

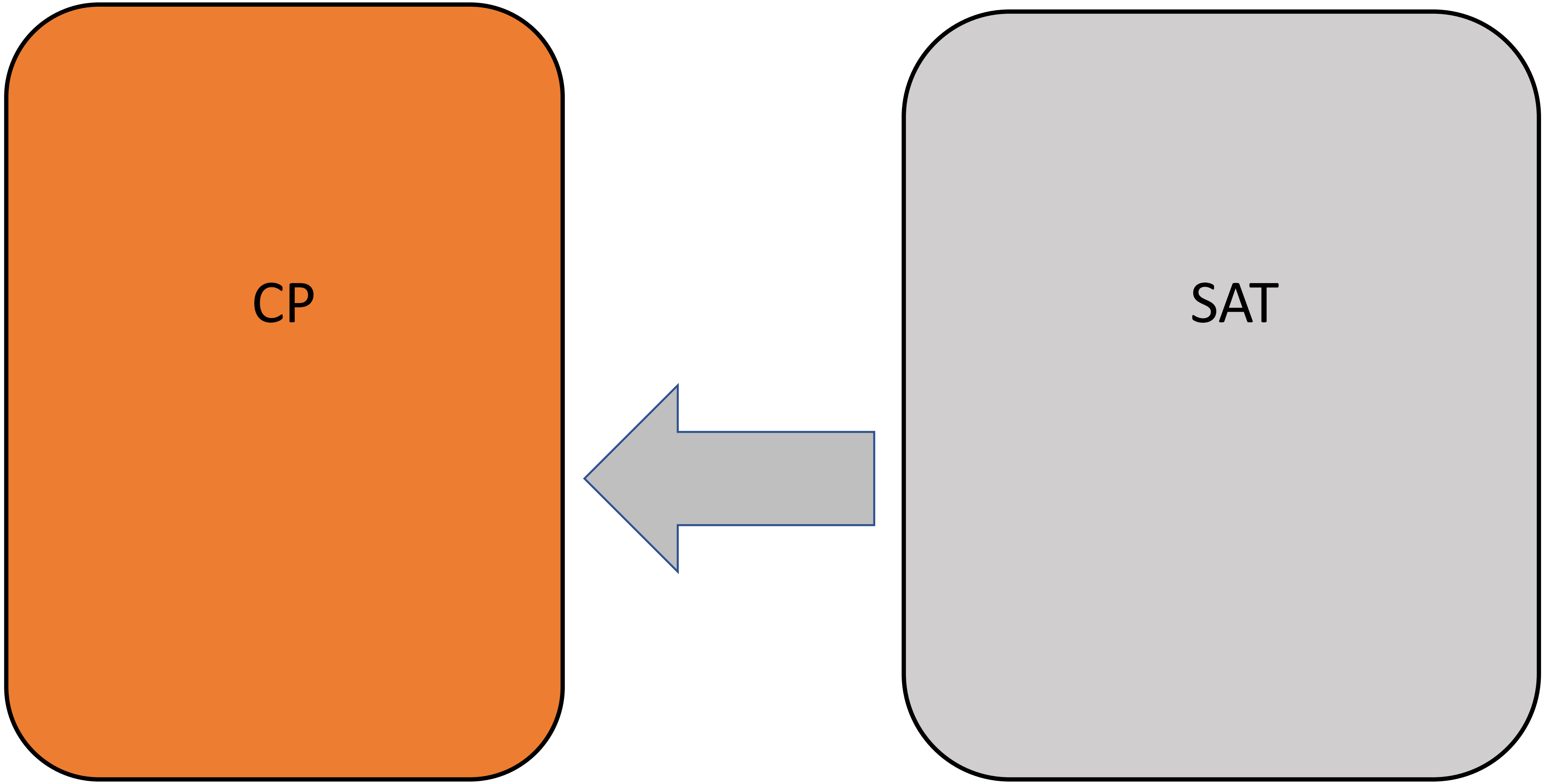
CP solvers best learning model



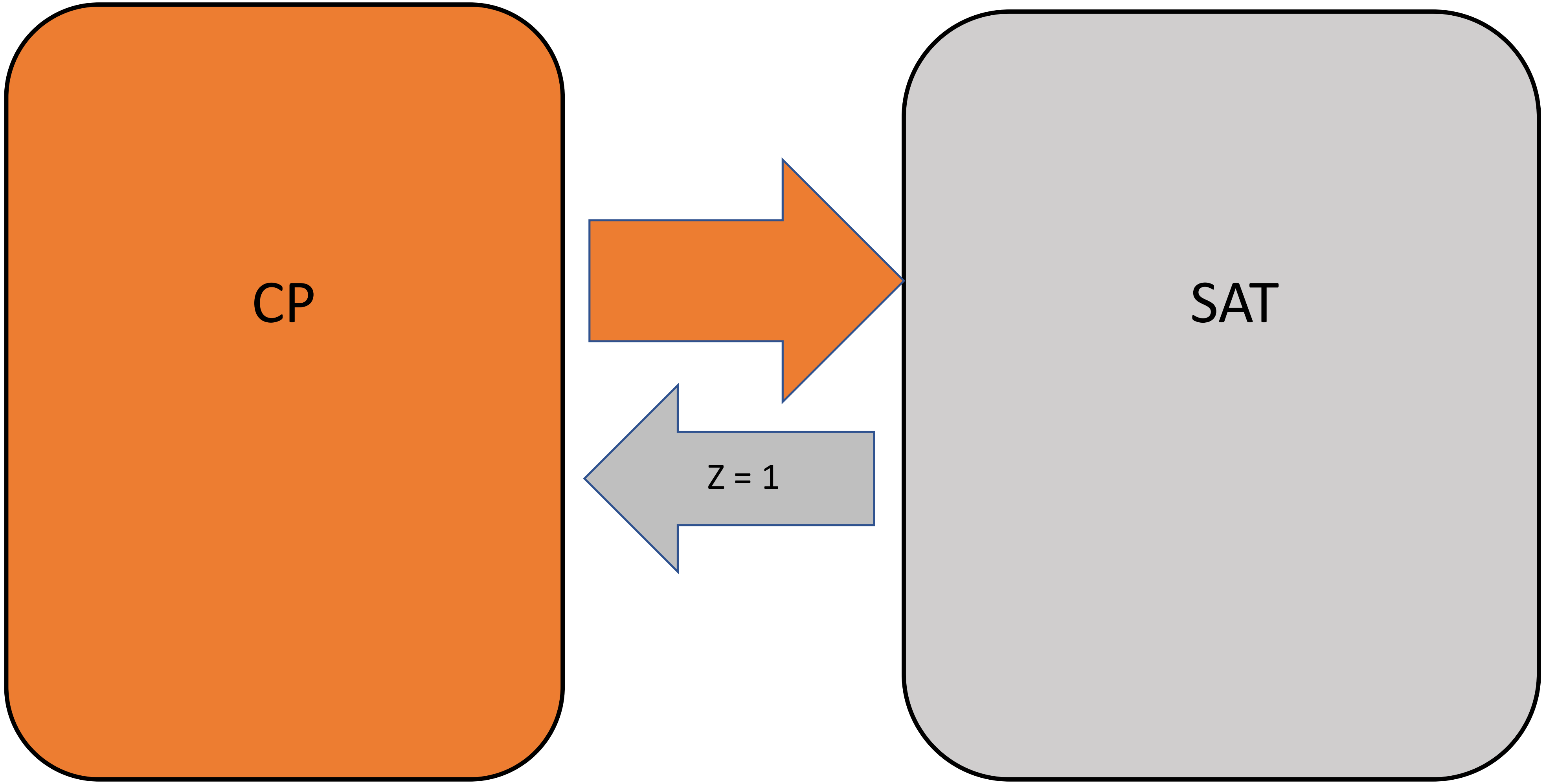
CP solvers best learning model



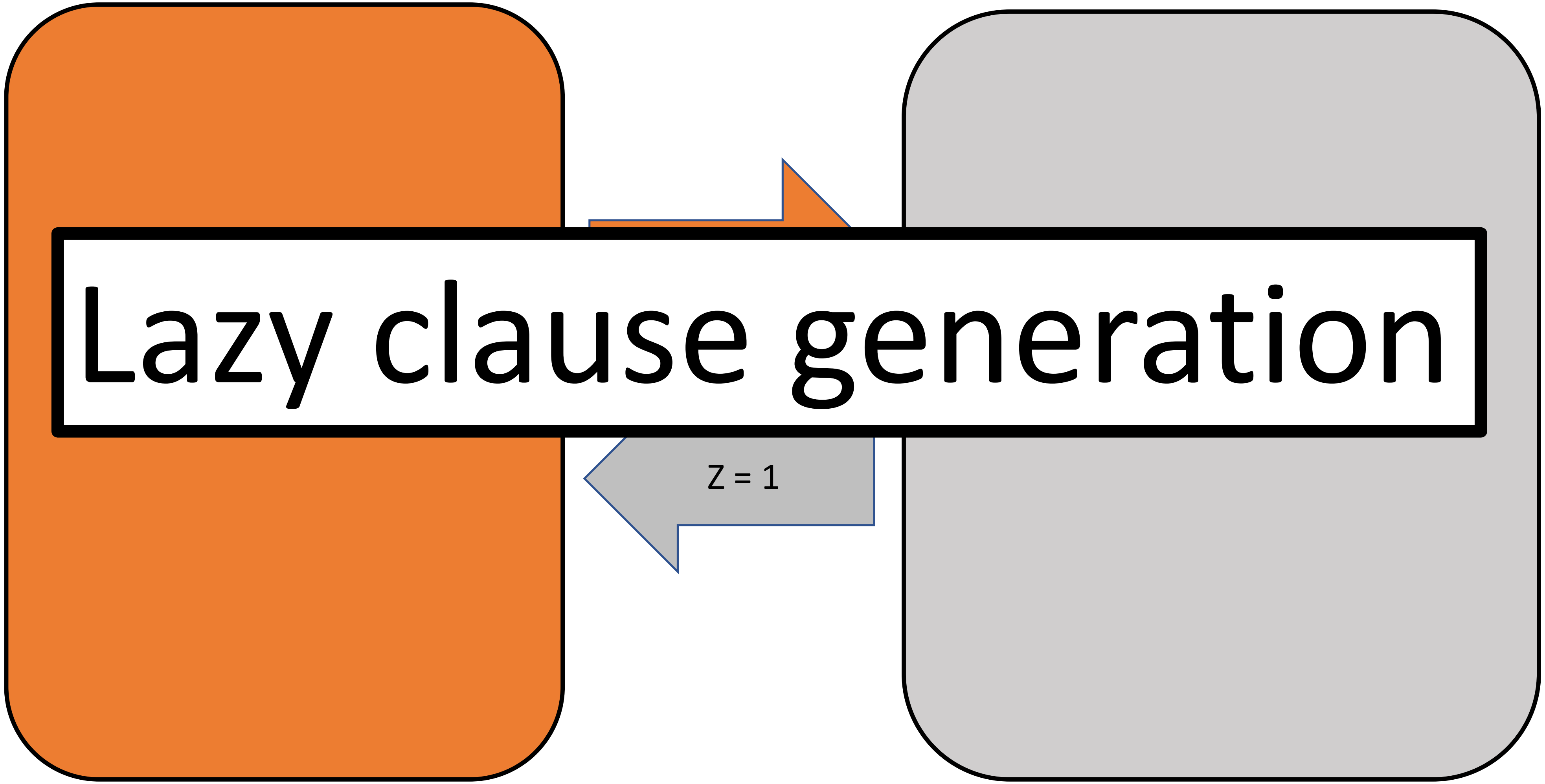
CP solvers best learning model



CP solvers best learning model



CP solvers best learning model



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

CSP   SMT   MIP   SAT

- 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

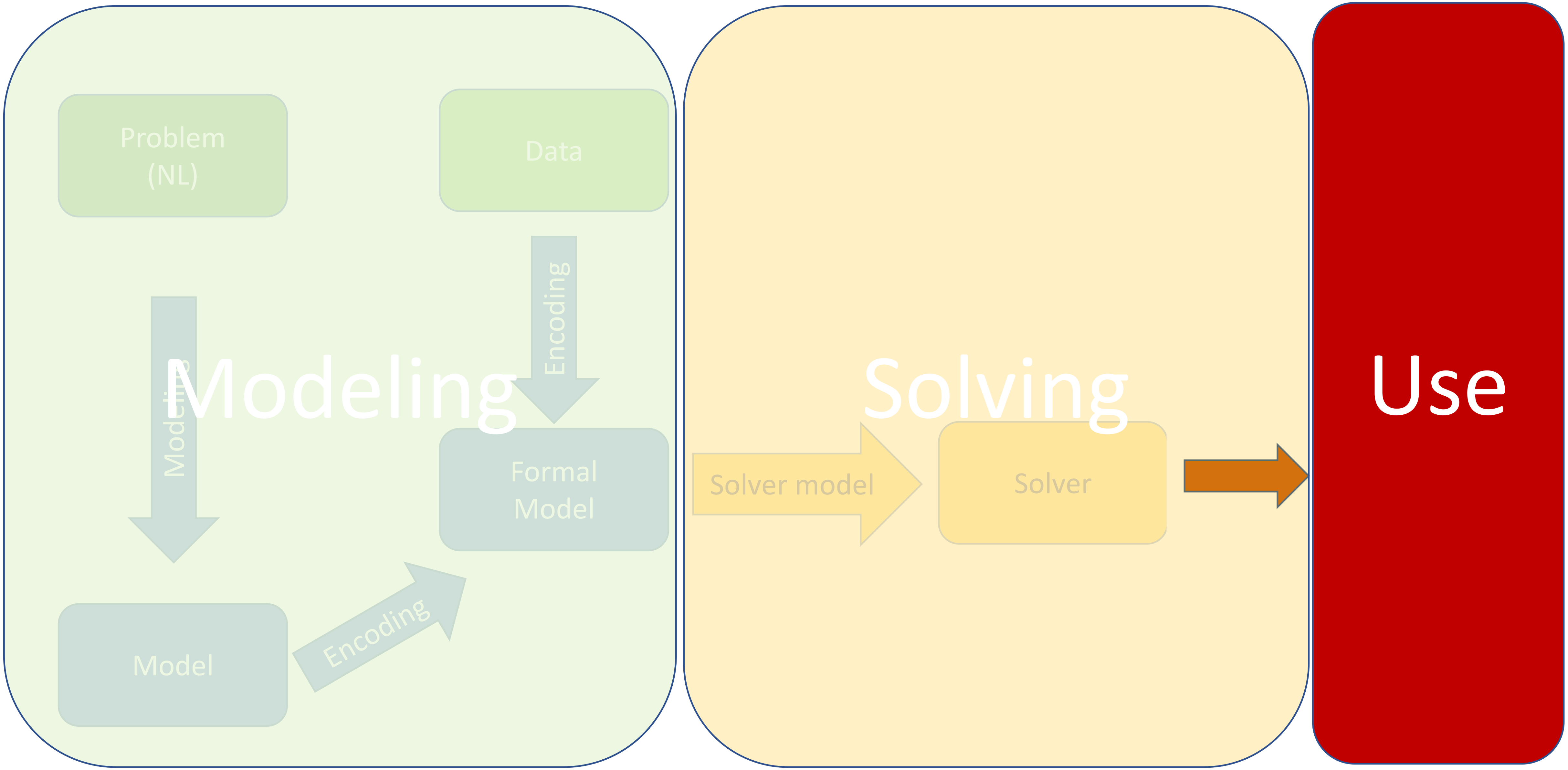
How solvers learn

CSP SMT MIP SAT

- 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

Overview


Problem
(NL)
Data
Modeling
Encoding
Modeling
Formal
Model
Model
Encoding
Solving
Solver model
Solver
Use

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

CSP

SMT

MIP

SAT

- 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)

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

SAT

(T/F)

$$(x_1 \vee \neg x_2) \wedge$$
$$(x_1 \vee \neg x_3) \wedge$$
$$\ldots$$

Solver independent modeling

Solvers modeling language

Minizinc

CSP

SMT

(Int/Real/Theory)

MIP

(Int/Real)

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

SAT

(T/F)

$$(x_1 \vee \neg x_2) \wedge$$
$$(x_1 \vee \neg x_3) \wedge$$
$$\ldots$$

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!