Introduction to Combinatorial Optimisation and Modelling

Emir Demirović

Algorithmics group | TU Delft

Algorithms for NP-Hard Problems (CSE2310 2023)

Lukina, Demirović, Yorke-Smith

Goal for Today

Combinatorial optimisation

Modelling

Modelling a scheduling problem

What is a combinatorial optimisation problem?

High School Timetabling

Coordinate teachers, rooms, and curriculum requirements subject to constraints and preferences

Demirović, Musliu; "MaxSAT-based large neighborhood search for high school timetabling“; Computers & Operations Research 78 (2017): 172-180.

Automotive Paintshop Scheduling

Car pieces need to be painted

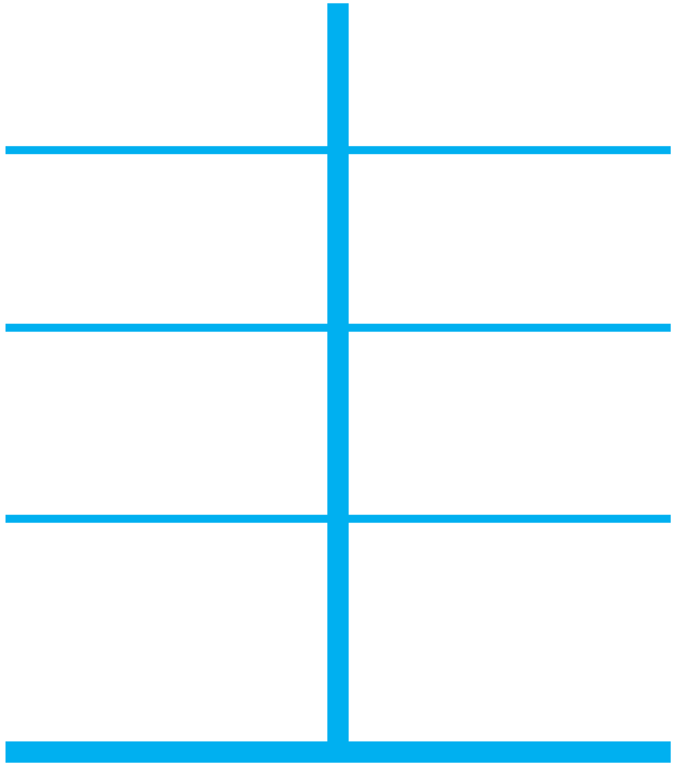
Sophisticated production process

Find the best schedule

Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

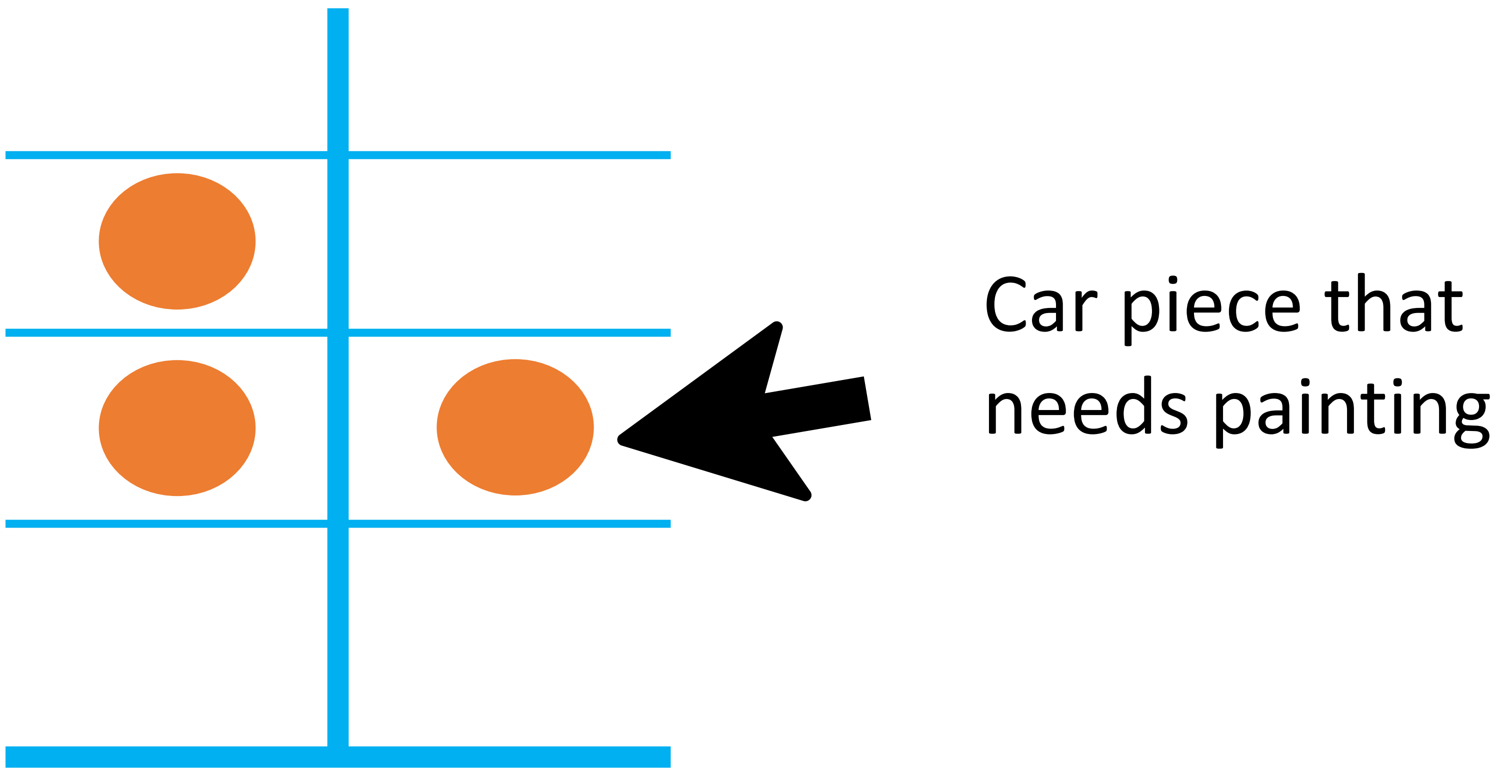
Automotive Paintshop Scheduling

Configurable carrier



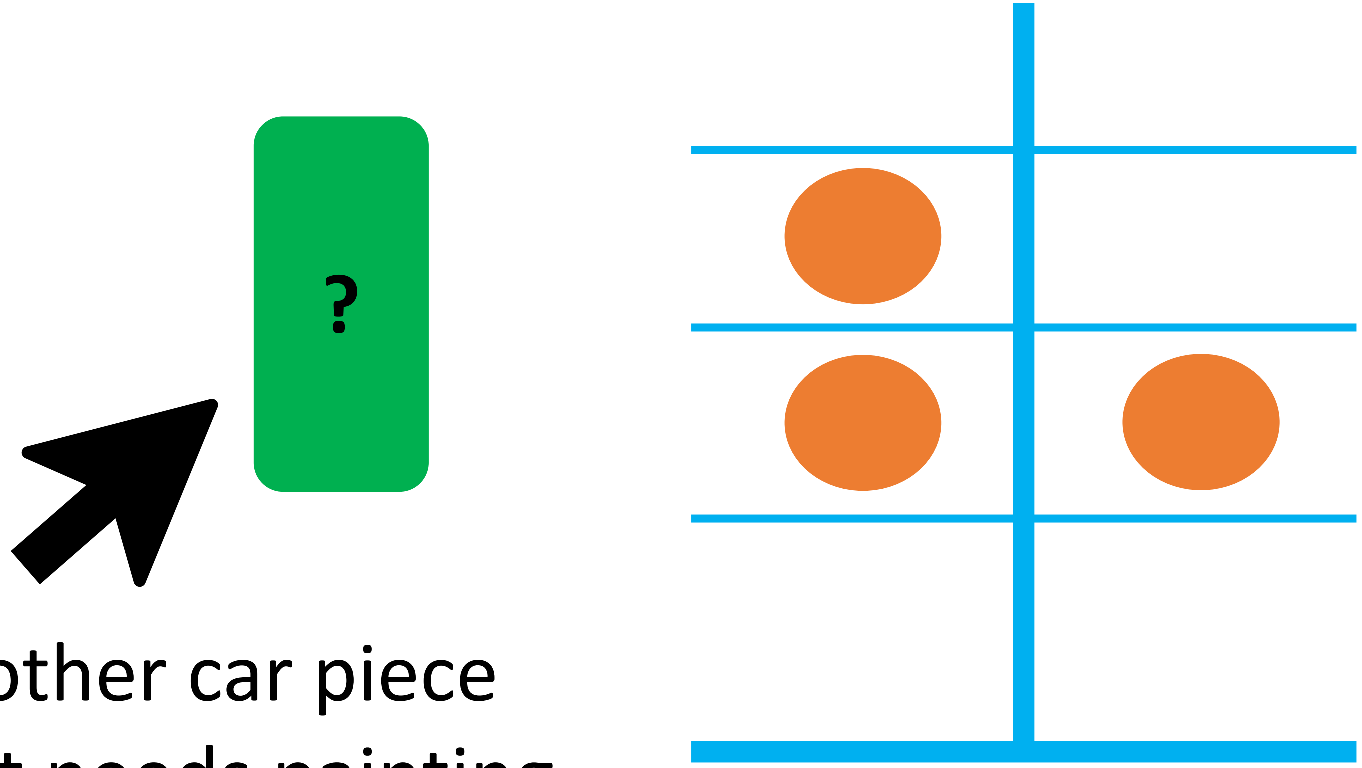
Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

Automotive Paintshop Scheduling



Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

Automotive Paintshop Scheduling

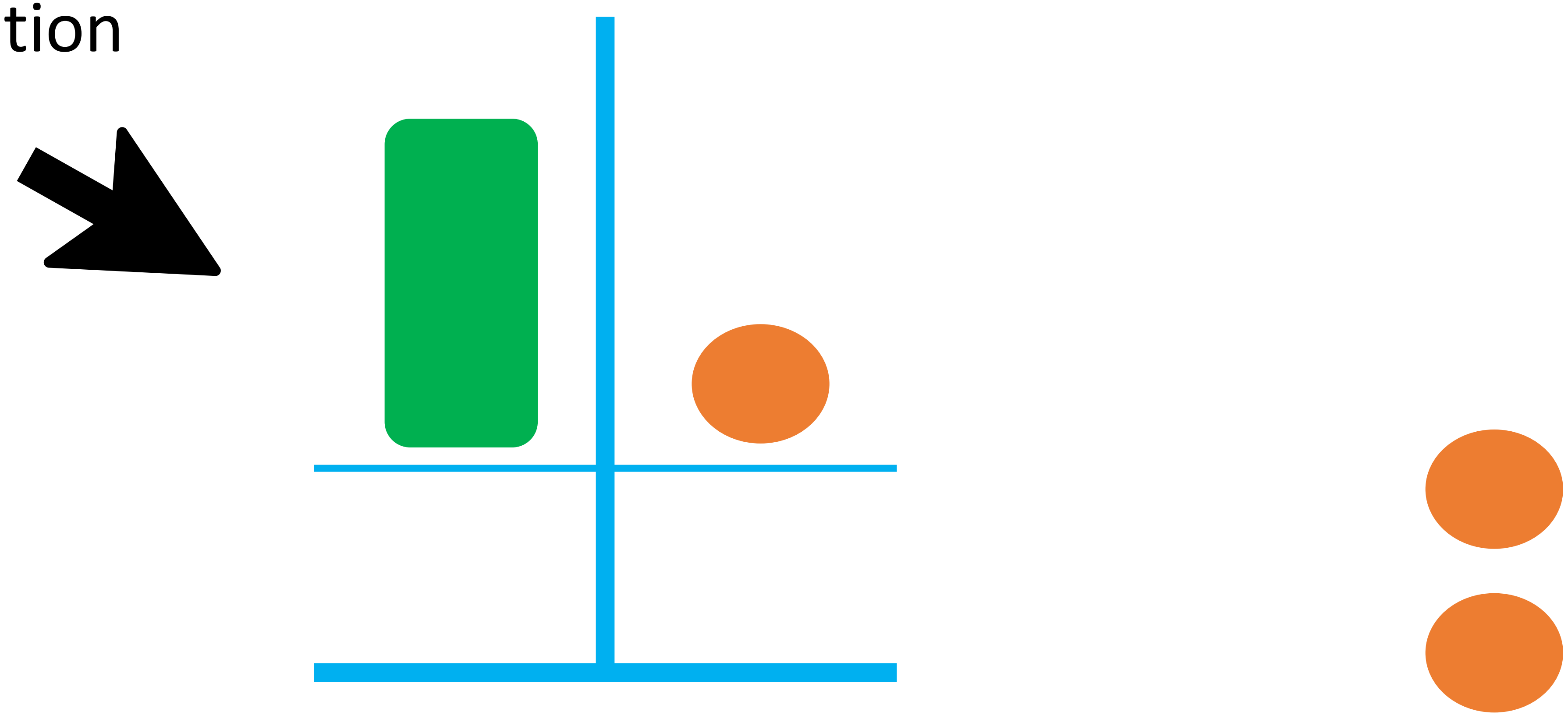


Another car piece
that needs painting

Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

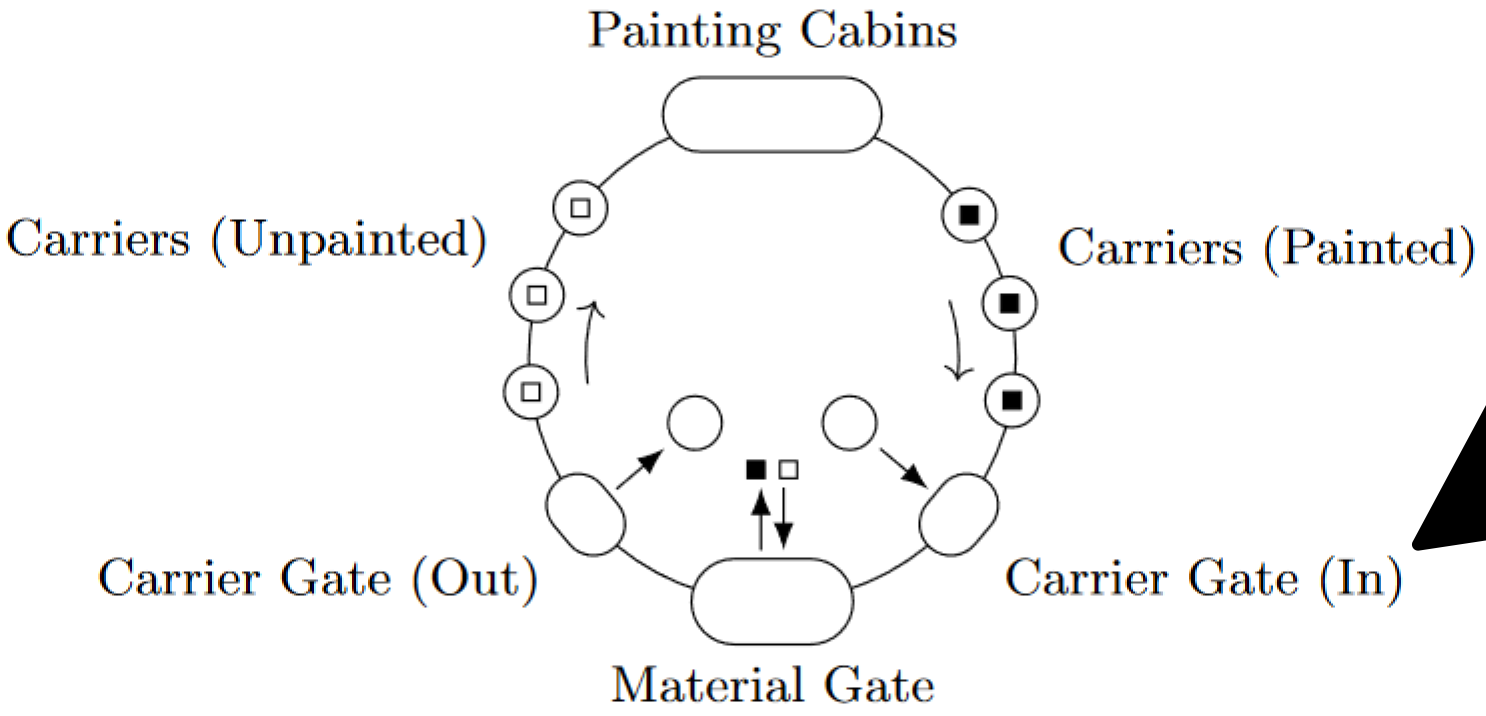
Automotive Paintshop Scheduling

New configuration



Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

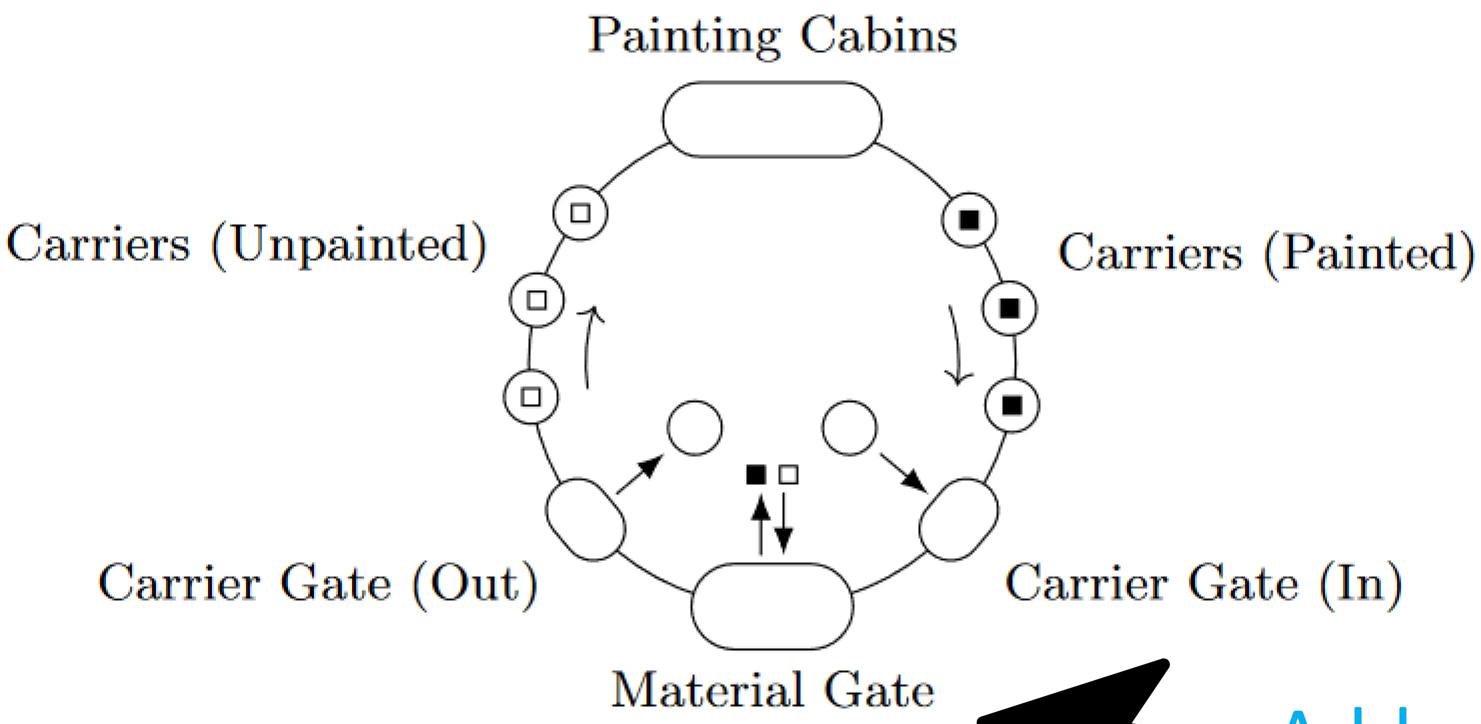
Automotive Paintshop Scheduling



Add new carriers

Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

Automotive Paintshop Scheduling

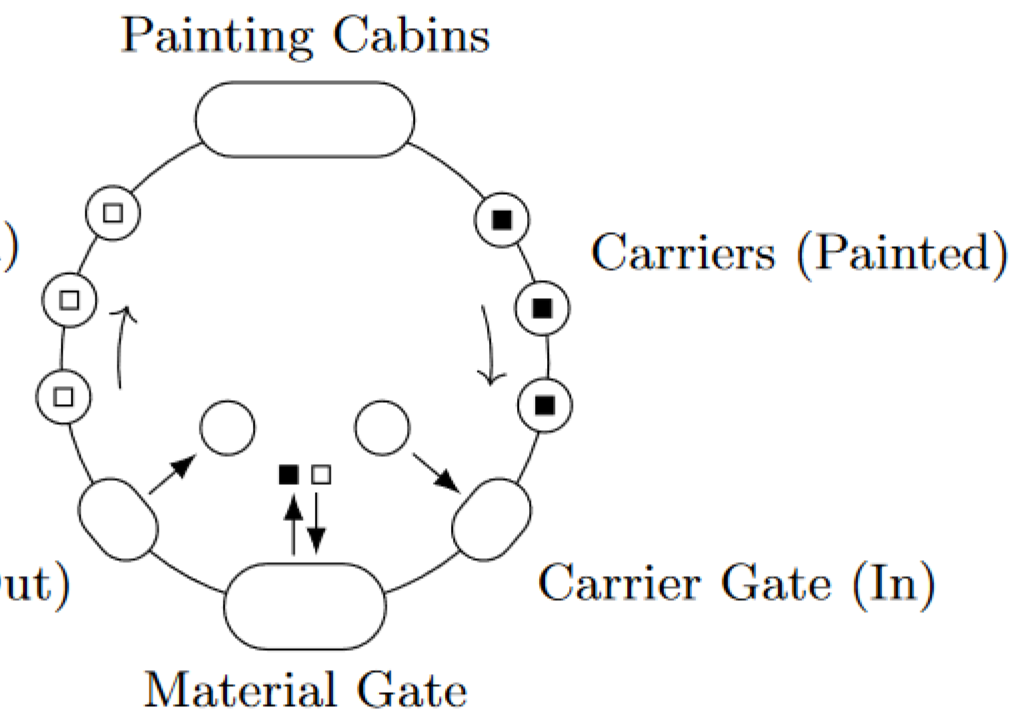


Add pieces on carriers

Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

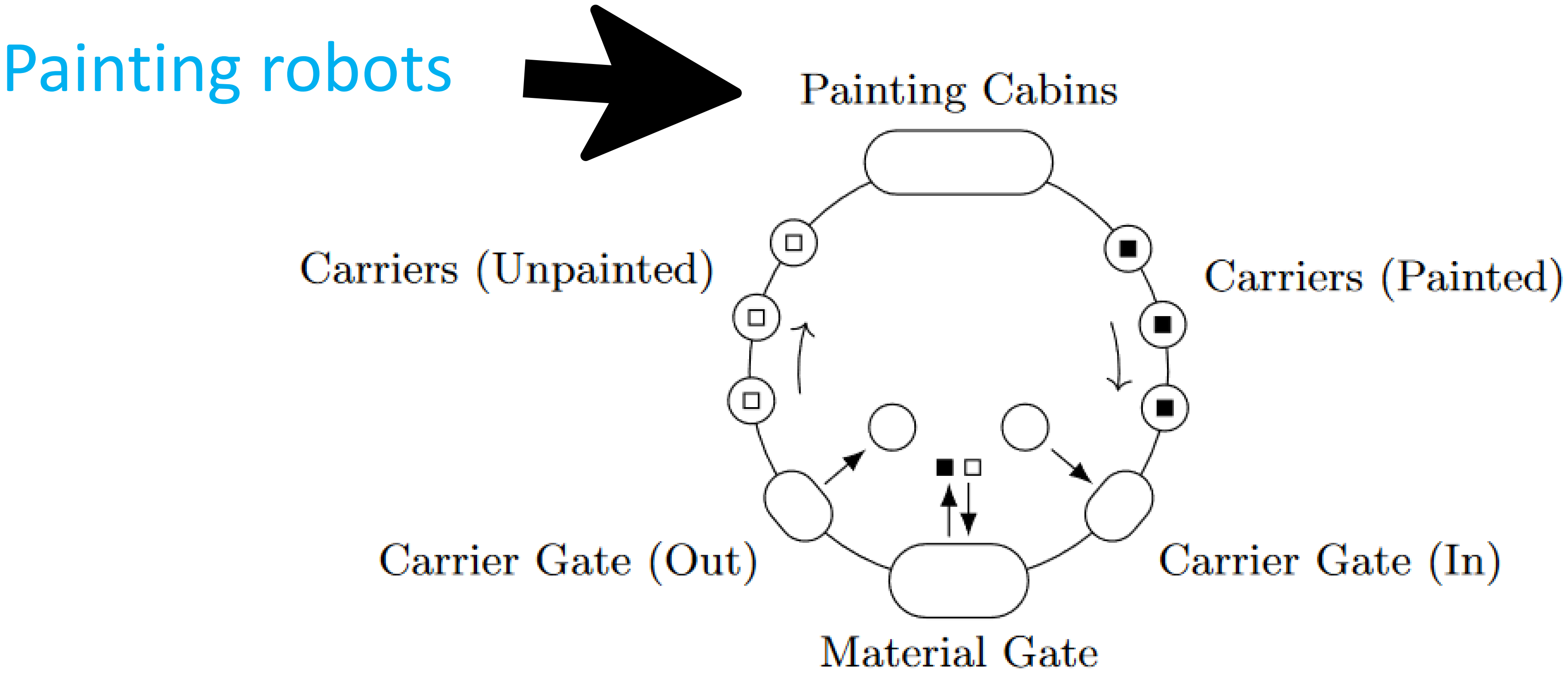
Automotive Paintshop Scheduling

Remove carriers



Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

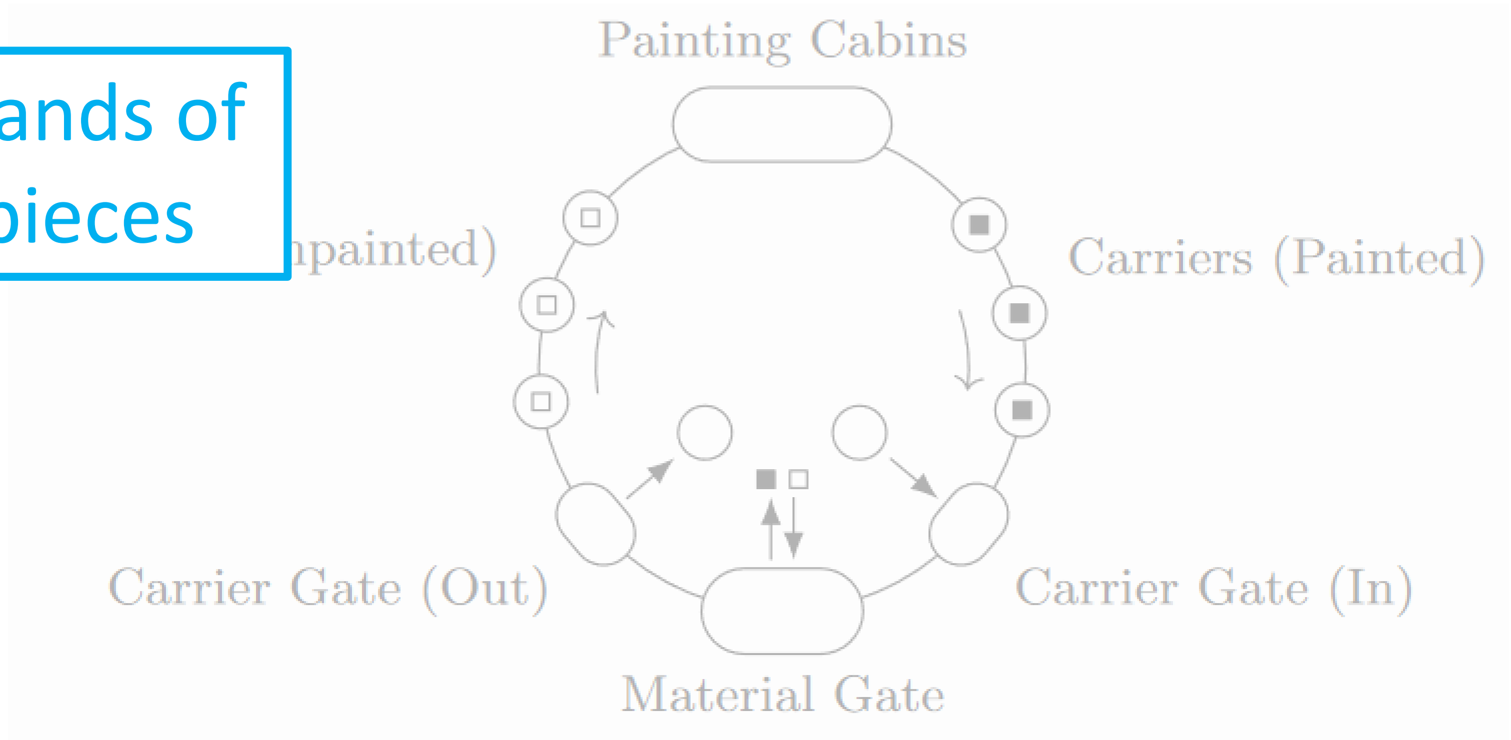
Automotive Paintshop Scheduling



Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

Automotive Paintshop Scheduling

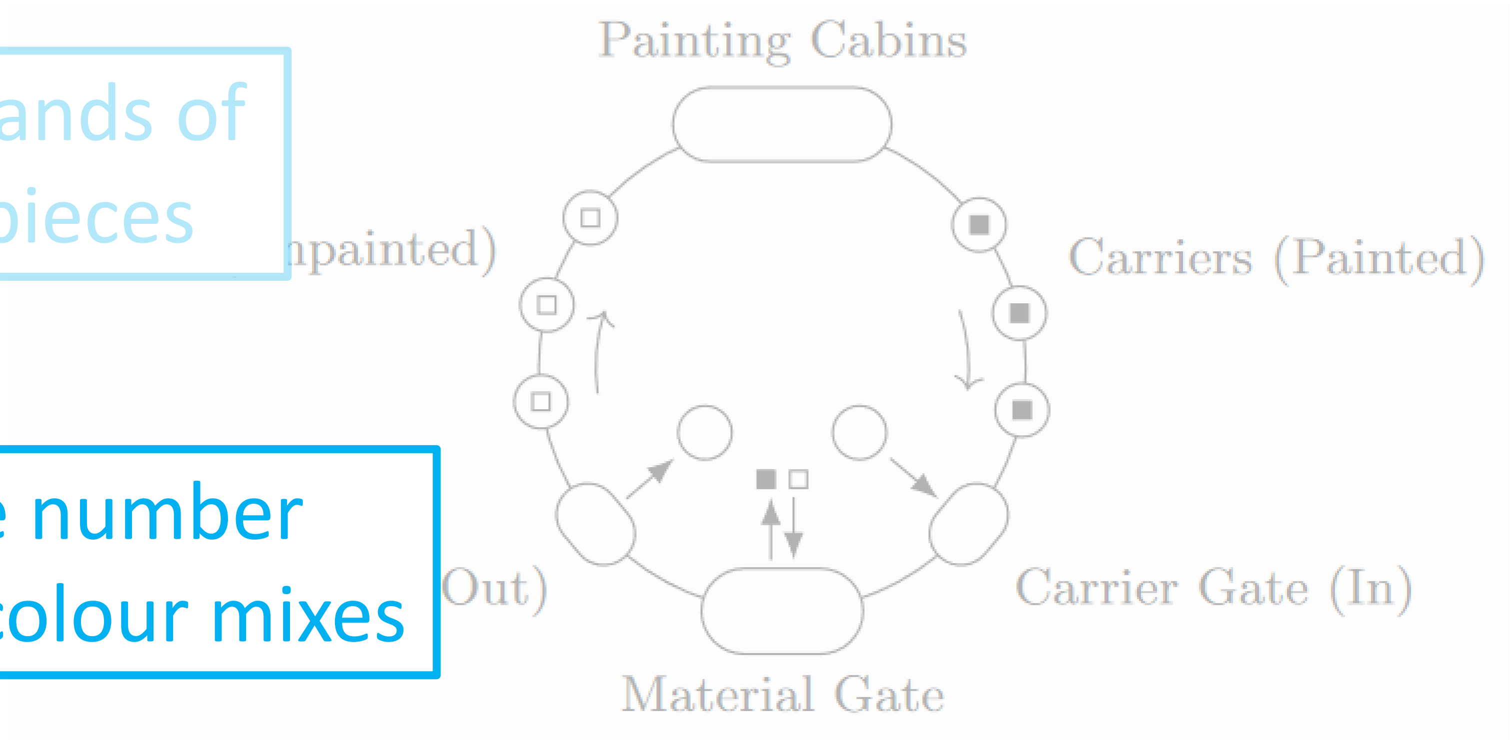
Thousands of car pieces



Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

Automotive Paintshop Scheduling

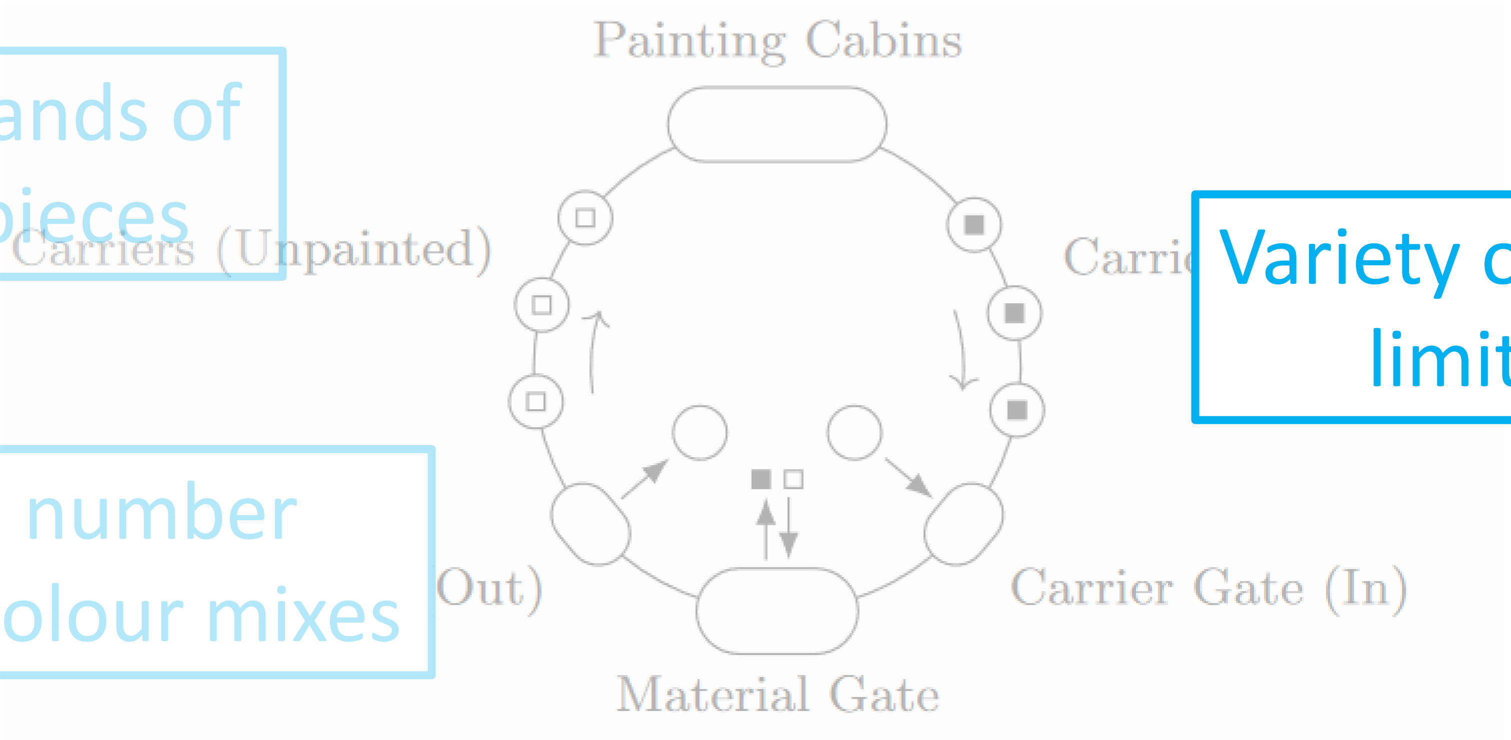
Thousands of car pieces



Incredible number of different colour mixes

Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

Automotive Paintshop Scheduling



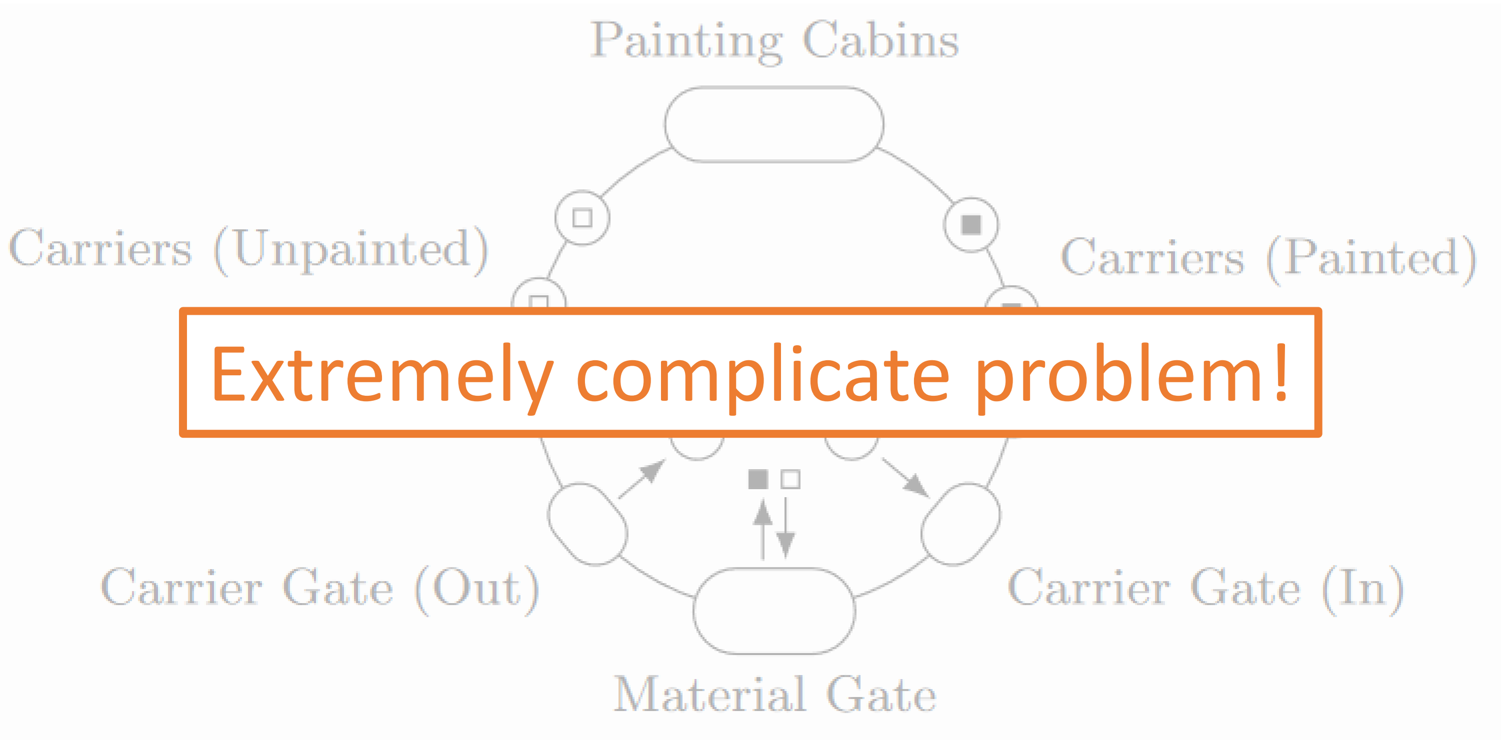
Thousands of car pieces

Variety of technical limitations

Incredible number of different colour mixes

Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

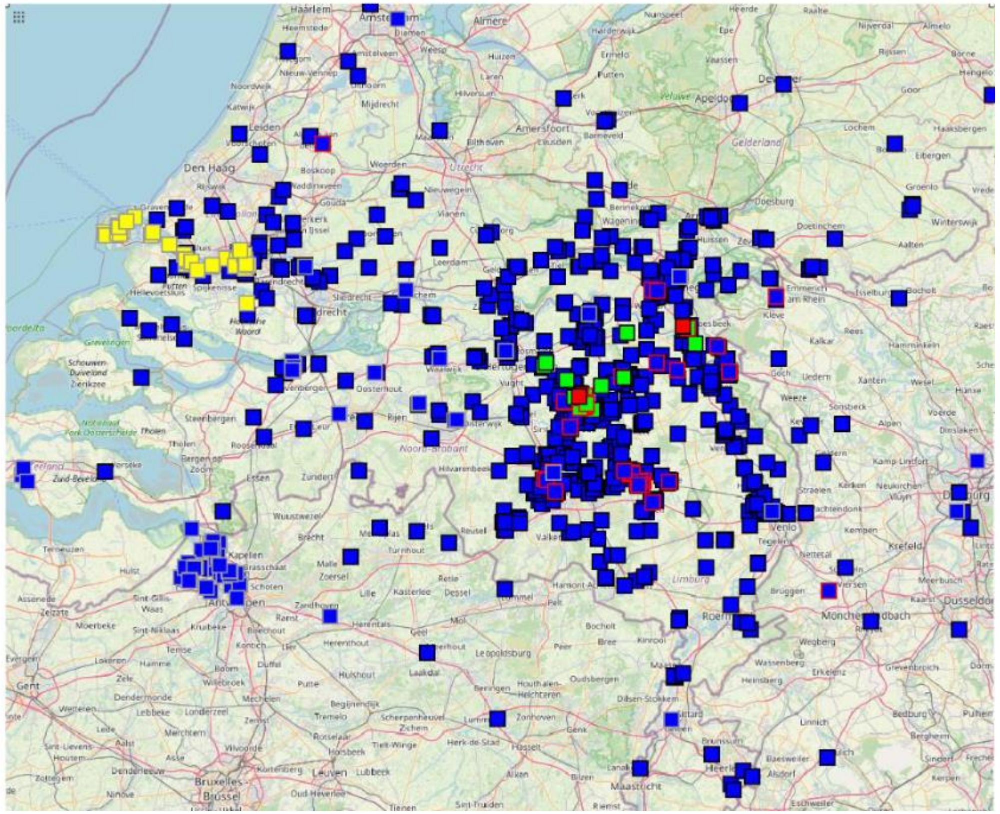
Automotive Paintshop Scheduling



Extremely complicate problem!

Winter, Musliu, Demirović, Mrkvicka; "Solution approaches for an automotive paint shop scheduling problem." *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019.

Truck Logistics

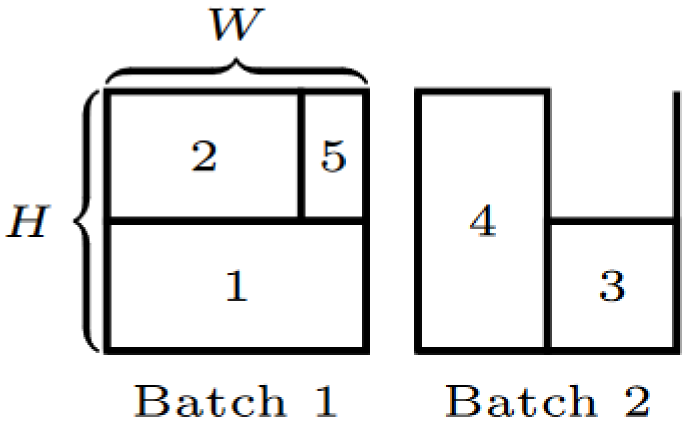


Plan truck routes
to deliver goods

Pingen, Van Ommeren, Van Leeuwen, Fransen, Elfrink, De Vries, Karunakaran, Demirović, Yorke-Smith
"Talking Trucks: Decentralized Collaborative Multi-Agent Order Scheduling for Self-Organizing Logistics„
Proceedings of the International Conference on Automated Planning and Scheduling. Vol. 32. 2022.

Picture taken from the master thesis of Yorick C. de Vries done at TU Delft in 2021

Batch Scheduling for Tool Coating



Tool Coating

Pack tools into batches

Schedule batches into ovens

Horn, Demirović, Yorke-Smith
"Parallel Batch Processing for the Coating Problem„
Proceedings of the International Conference on Automated Planning and Scheduling (to appear). 2023.

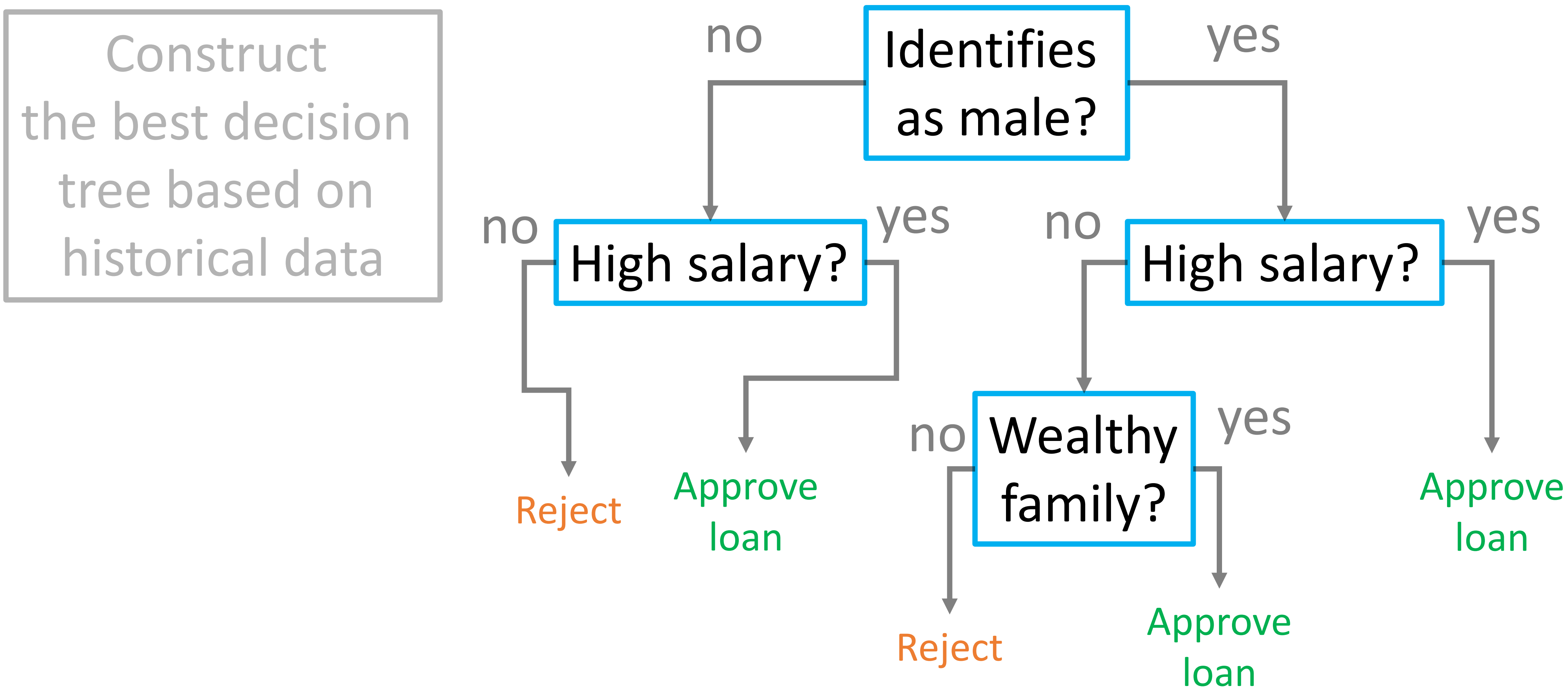
Fair and Optimal Decision Trees (Machine Learning)

Construct
the best decision
tree based on
historical data

"automate the bank loan approval process"

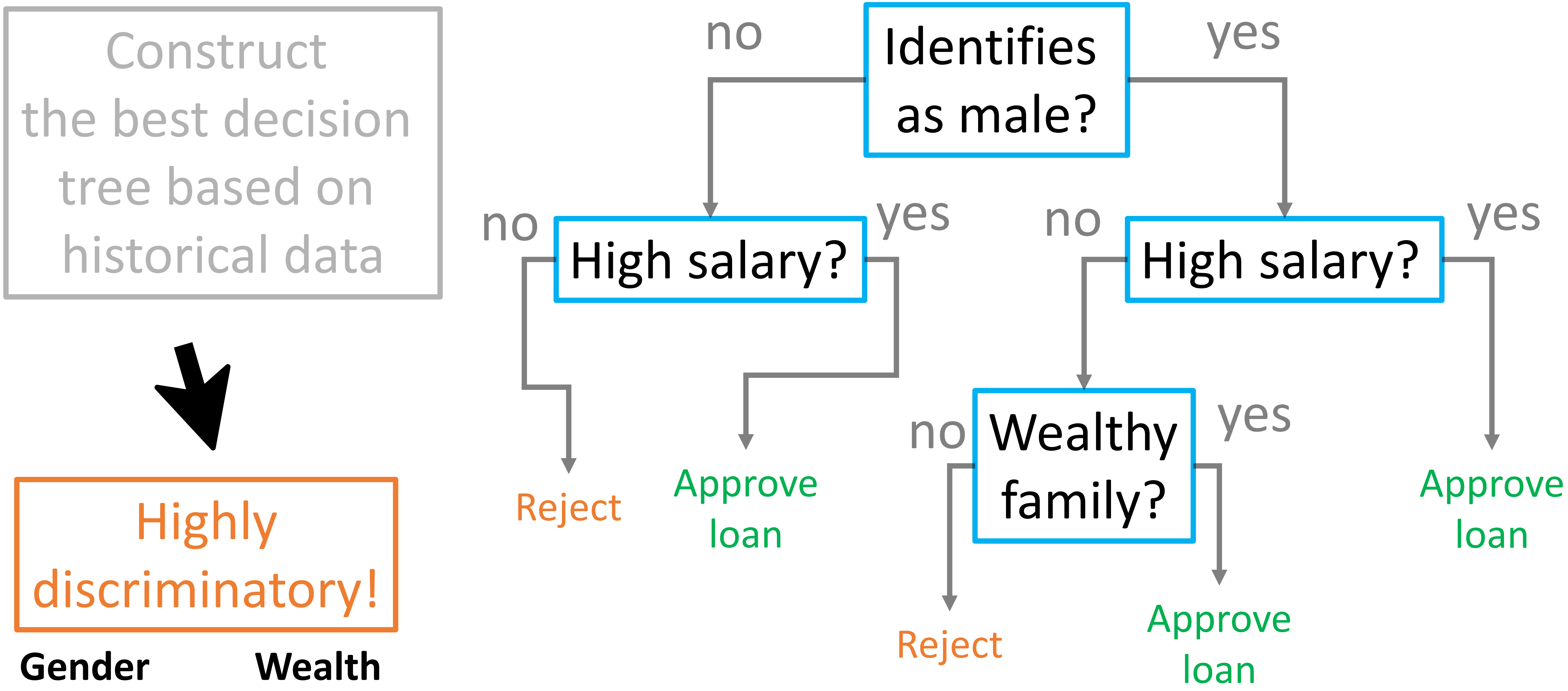
Van Der Linden, De Weerdt, Demirović; "Fair and Optimal Decision Trees: A Dynamic Programming Approach." Advances in Neural Information Processing Systems. 2022.

Fair and Optimal Decision Trees (Machine Learning)



Van Der Linden, De Weerdt, Demirović; "Fair and Optimal Decision Trees: A Dynamic Programming Approach." Advances in Neural Information Processing Systems. 2022.

Fair and Optimal Decision Trees (Machine Learning)



Van Der Linden, De Weerdt, Demirović; "Fair and Optimal Decision Trees: A Dynamic Programming Approach." Advances in Neural Information Processing Systems. 2022.

Fair and Optimal Decision Trees (Machine Learning)

Construct
the best decision
tree based on
historical data...

Difficult
combinatorial
problem!

...that is also fair!

Van Der Linden, De Weerdt, Demirović; "Fair and Optimal Decision Trees: A Dynamic Programming Approach." Advances in Neural Information Processing Systems. 2022.

High School Timetabling

Automotive Paint Shop Scheduling

Truck Logistics

Batch Scheduling for Coating Tools

Fair and Optimal Decision Trees

...all examples of combinatorial optimisation problems!

Combinatorial Optimisation Problem

$$\boldsymbol{X} \in \boldsymbol{C} \subseteq \mathbb{N}^{\boldsymbol{n}}$$

Solution

Combinatorial Optimisation Problem

$$\boldsymbol{X} \in \boldsymbol{C} \subseteq \mathbb{N}^{\boldsymbol{n}}$$

Set of feasible solutions, implicitly defined through constraints

Combinatorial Optimisation Problem

Objective function

$$\boldsymbol{min}\ \boldsymbol{F}(\boldsymbol{X})$$
$$\boldsymbol{X} \in \boldsymbol{C} \subseteq \mathbb{N}^{\boldsymbol{n}}$$

Combinatorial Optimisation Problem

$$\boldsymbol{min\ F(X)}$$
$$\boldsymbol{X \in C \subseteq \mathbb{N}^n}$$

Combinatorial Optimisation Problem

$$\min F(X)$$
$$X \in C \subseteq \mathbb{N}^n$$

How to efficiently utilise resources?

Combinatorial optimisation is everywhere!

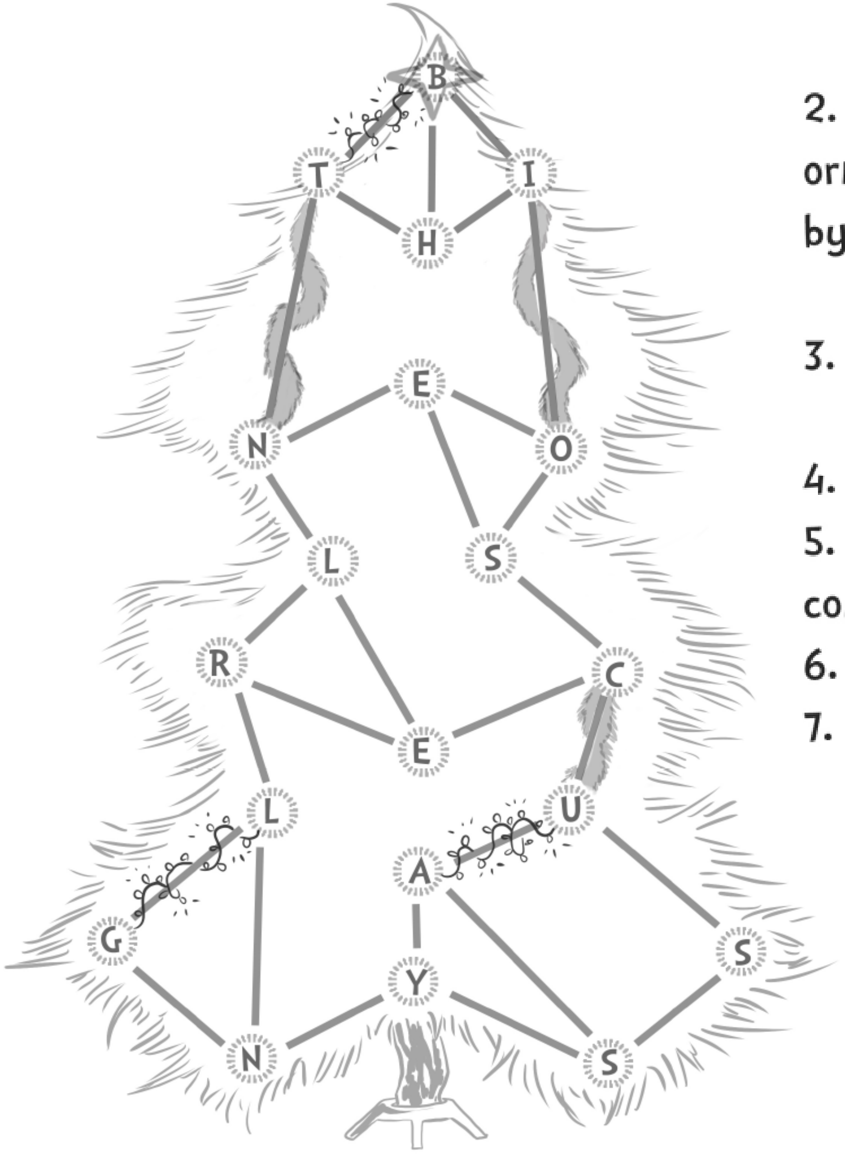
How to solve combinatorial problems?

Step 1: “Modelling” → Define the problem mathematically

Essential

Difficult

Not uncommon to spend most time on modelling!



2. Garlands always connect two of the same type of ornament. Conversely, any ornament which is connected to another ornament of the same type MUST be connected by a garland.

3. Lights are always connected to a Circle at one or the other end.

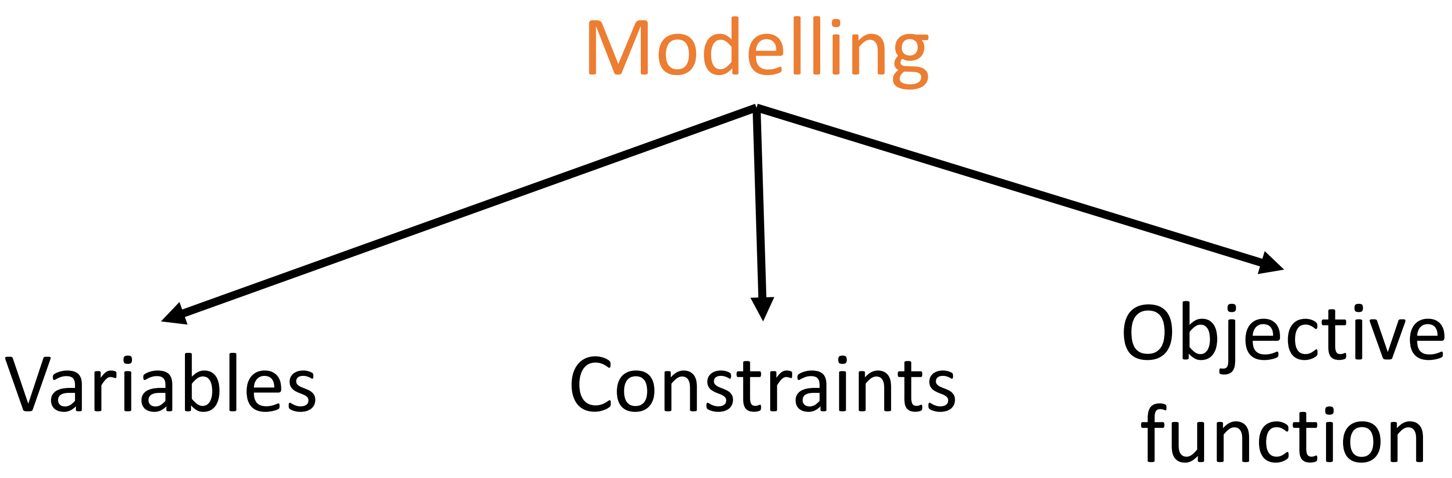
4. Every Circle is directly connected to at least one Star.
5. Every Diamond must be connected to a Cane lower than it, and every Cane must be connected to a Diamond higher than it.
6. Canes are *never* connected to each other.
7. Stars can only be placed at spots connected to all three of the other ornament types.

Model the puzzle!

Puzzle taken from Puzzled Pint Dec'20

“All models are wrong, but some are useful”

-George Box


Modelling
Variables
Constraints
Objective function

Modelling

Variables

Constraints

Objective function

"decisions we can make"

e.g., $X_{N,circle} \in \{0,1\}$

"is circle placed on node N?"

Modelling

Variables

Constraints

Objective function

"feasibility"

e.g., "every circle is connected to a star"

Modelling

Variables

Constraints

Objective function

“quality”

“soft constraints”

e.g., use the least number of circles

Modelling

Variables

Constraints

Objective function

"quality"

"soft constraints"

$$\boldsymbol{min\ F(X)}$$
$$\boldsymbol{X \in C \subseteq \mathbb{N}^n}$$

Modelling

Variables

Constraints

Objective function

“quality”

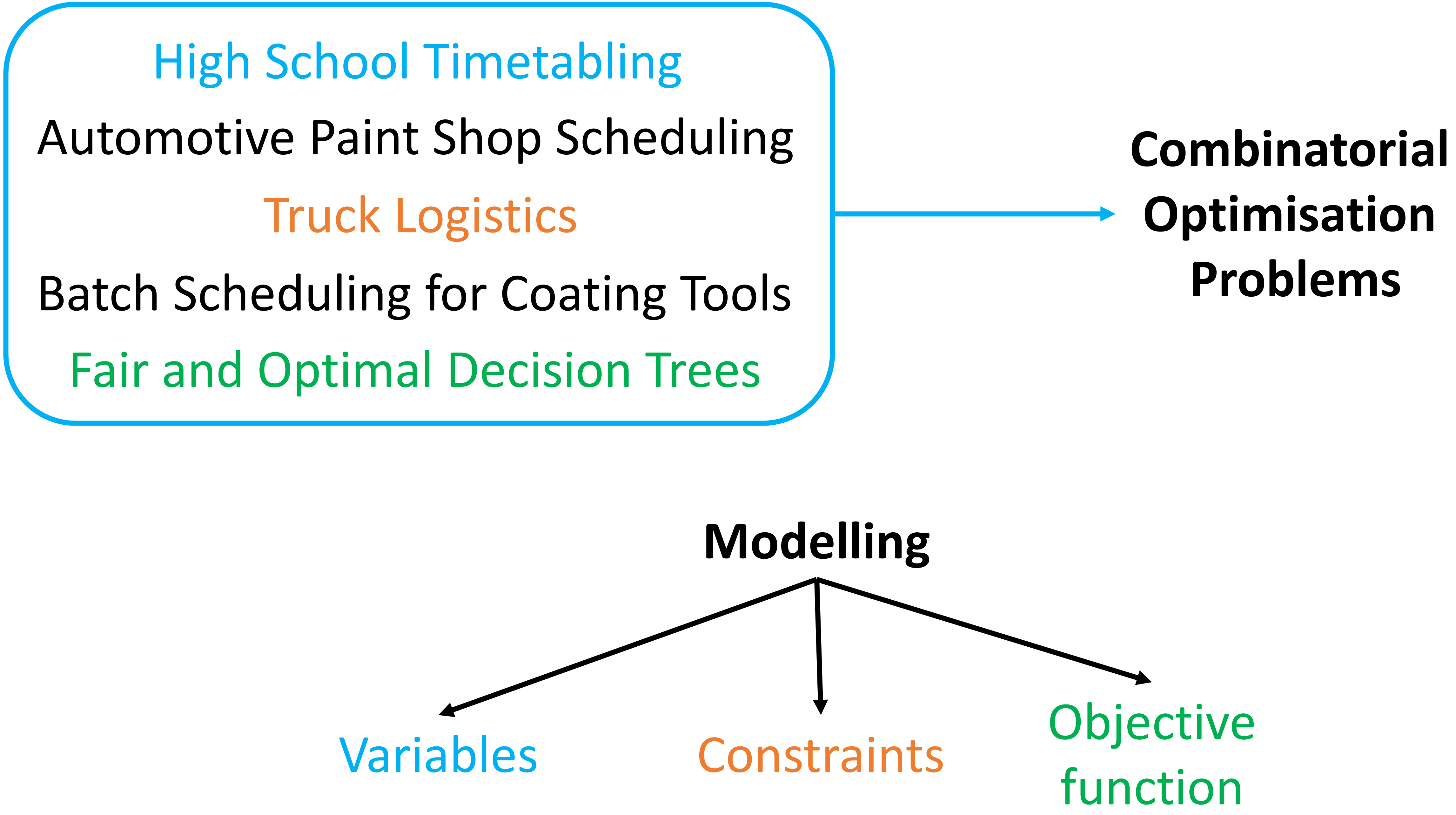
“soft constraints”

$$\boldsymbol{min\ F(X)}$$
$$\boldsymbol{X \in C} \subseteq \mathbb{N}^{\boldsymbol{n}}$$

Optimal solution X^*

$$\forall \boldsymbol{X} \in \boldsymbol{C}\colon\ \boldsymbol{F}(\boldsymbol{X}^*) \leq \boldsymbol{F}(\boldsymbol{X})$$

Cannot be better than the optimal solution!

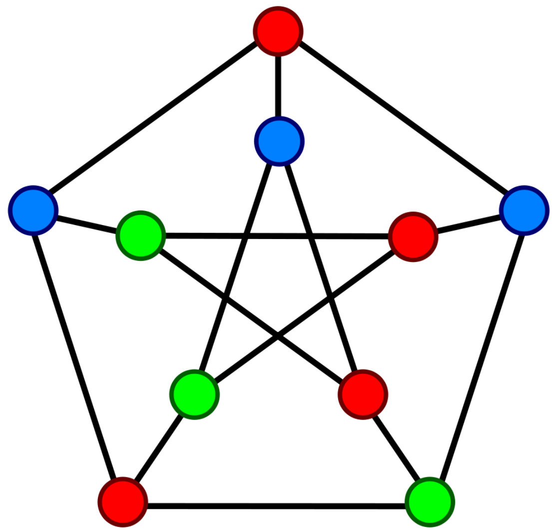


High School Timetabling
Automotive Paint Shop Scheduling
Truck Logistics
Batch Scheduling for Coating Tools
Fair and Optimal Decision Trees
Combinatorial Optimisation Problems
Modelling
Variables
Constraints
Objective function

Modelling Example: Graph Colouring

Each node is assigned a colour

No two neighbours may share the same colour



Model the problem!

Variables

Constraints

Objective function

Modelling Example: High School Timetabling

Each course is assigned a lecturer and a group of students

Each course lasts for one hour

Lecturer and students can only attend one course at any given time

Assume there are enough rooms

Goal: find a timetable without clashes

Model the problem!

Variables

Constraints

Objective function

Many different problems share (similar) models!

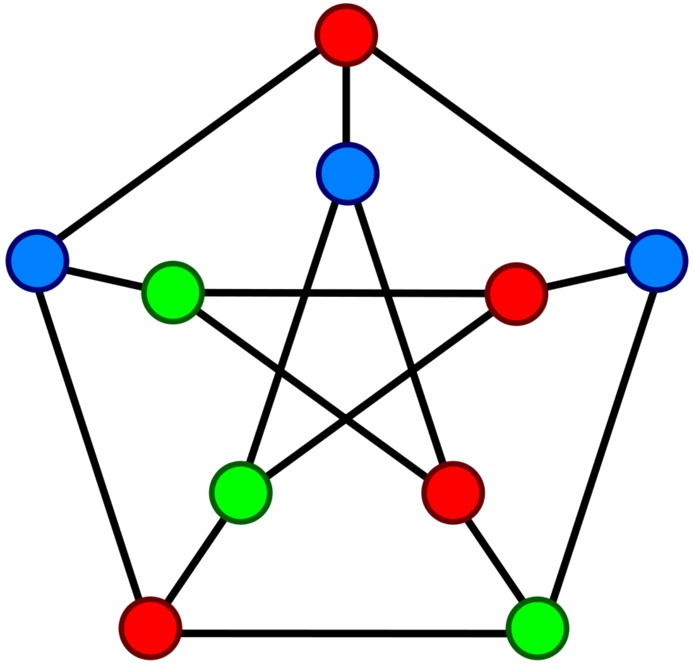
Many different problems share (similar) models!

Timetabling and graph colouring

Courses → nodes

Incompatibilities → edges

Time slots → colours



Resource-Constrained Project Scheduling Problem

$$task_i = (s_i, D_i, R_i)$$

Integer variable representing possible starting times

Resource-Constrained Project Scheduling Problem

$$task_i = (s_i, D_i, R_i)$$

Constant representing task duration

Resource-Constrained Project Scheduling Problem

$$task_i = (s_i, D_i, R_i)$$

Constant representing resource requirements

Resource-Constrained Project Scheduling Problem

$$task_i = (s_i, D_i, R_i)$$

$$R_{max}$$

Constant representing maximum number of available resources

Resource-Constrained Project Scheduling Problem

$$task_i = (s_i, D_i, R_i)$$

$$R_{max}$$

“The resource consumption cannot exceed the maximum at any time”

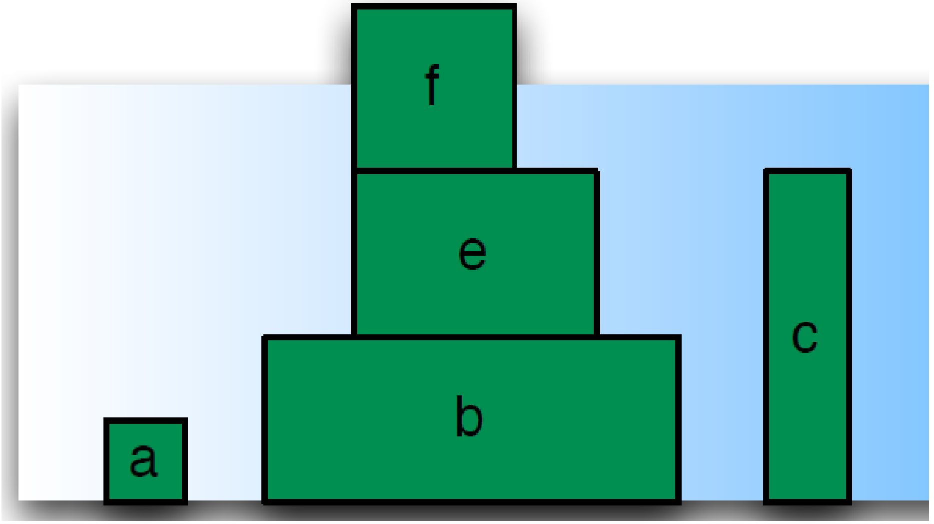
Resource-Constrained Project Scheduling Problem

$$task_i = (s_i, D_i, R_i)$$

$$R_{max}$$

“The resource consumption cannot exceed the maximum at any time”

Infeasible!



Picture taken from the PhD thesis of Andreas Schutt (2011)

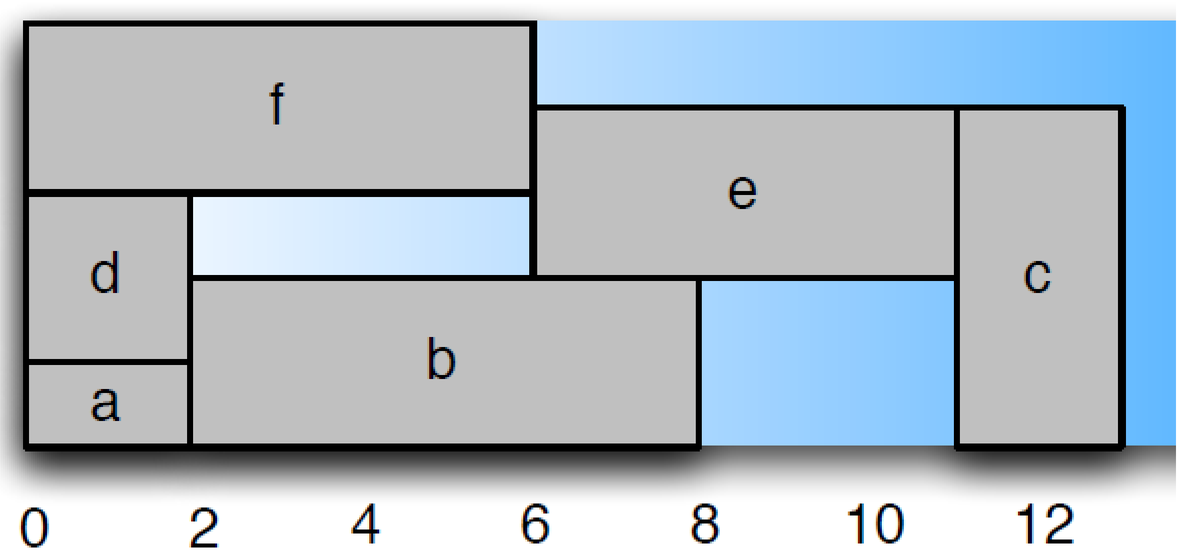
Resource-Constrained Project Scheduling Problem

$$task_i = (s_i, D_i, R_i)$$

$$R_{max}$$

"The resource consumption cannot exceed the maximum at any time"

Feasible!



Picture taken from the PhD thesis of Andreas Schutt (2011)

Resource-Constrained Project Scheduling Problem

Precedence Relation Constraints

$$task_a < task_b$$

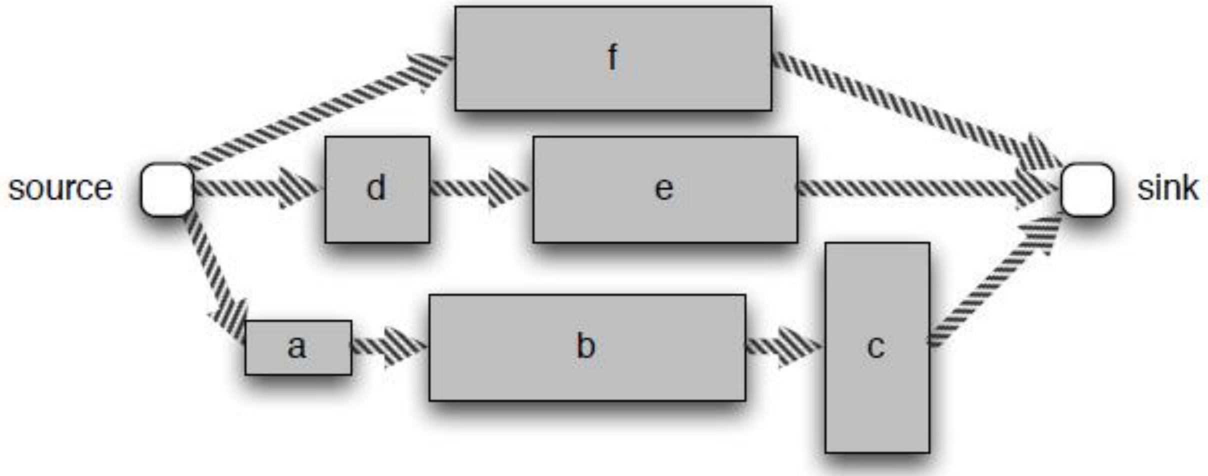
$$task_d < task_e$$

Resource-Constrained Project Scheduling Problem

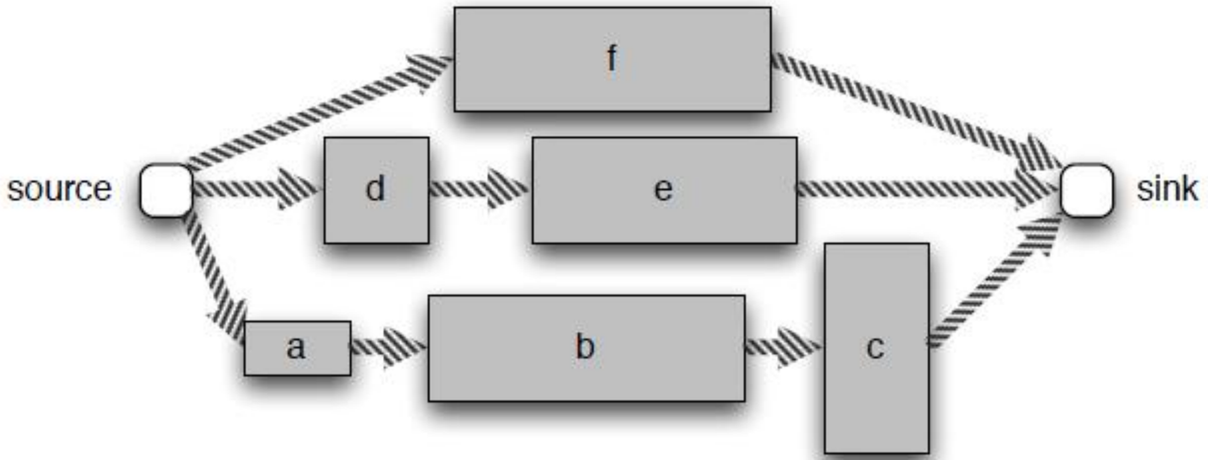
Precedence Relation Constraints

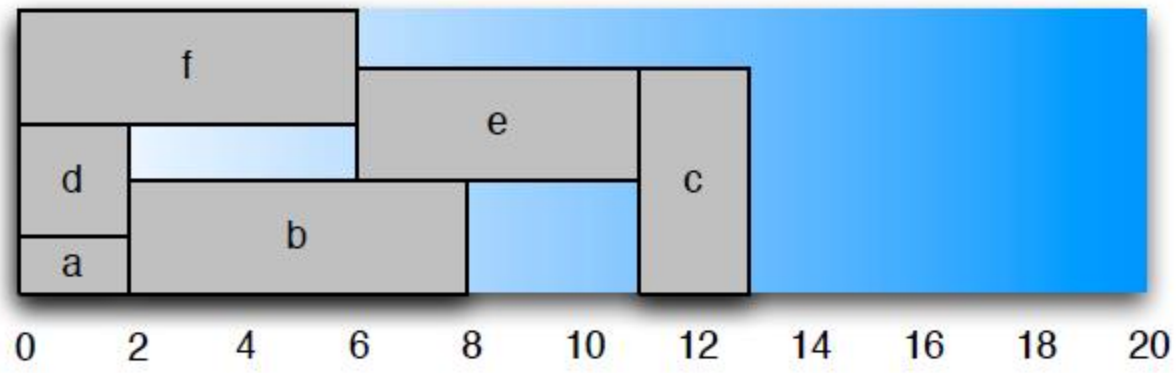
$$task_a < task_b$$

$$task_d < task_e$$



Resource-Constrained Project Scheduling Problem





Constraints

Precedence Relations

$$task_i < task_j$$

Resource Limitation

$$task_i = (s_i, D_i, R_i)$$

Model the problem using Boolean variables

Image taken "Improving Scheduling by Learning", Andreas Schutt, PhD thesis, University of Melbourne, 2011

$\boxed{\text{Modelling is the first step to solving any problem}}$

Precise mathematical formulation

Summary

Combinatorial optimisation

Modelling

Timetabling

Graph Colouring

Scheduling

Next time:
algorithms to find solutions to our models!

$X_{Node, shape} \in \{0, 1\}$

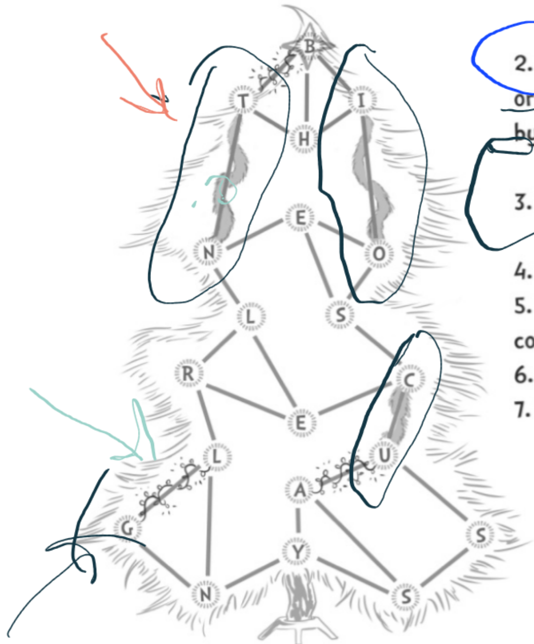
"shape is at the node"

$X_{B, star} = 1 \rightarrow$ there is a star at node B

$X_{T, circle}$

$X_{T, shape} = X_{N, shape}$

$X_{T, circle} \leftrightarrow X_{N, circle}$



2. Garlands always connect two of the same type of ornament. Conversely, any ornament which is connected to another ornament of the same type MUST be connected by a garland.

3. Lights are always connected to a Circle at one or the other end.

4. Every Circle is directly connected to at least one Star.
5. Every Diamond must be connected to a Cane lower than it, and every Cane must be connected to a Diamond higher than it.
6. Canes are *never* connected to each other.
7. Stars can only be placed at spots connected to all three of the other ornament types.

Model the puzzle!

Puzzle taken from Puzzled Pint Dec'20

$(X_{G, circle} \vee X_{L, circle}) \leftarrow ?,$

$$X_{G,\,\text{cirle}} \twoheadrightarrow X_{L,\,\text{circle}}$$

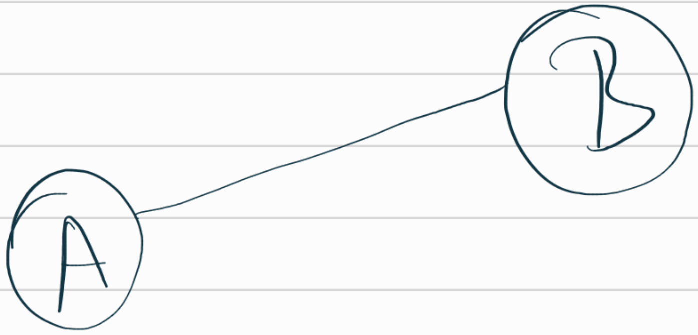
$$X_{L,\,\text{circle}} \hookrightarrow \overline{X_{G,\,\text{circle}}}$$

$$\Bigg\} \; 2$$

$$\longrightarrow \qquad \longleftrightarrow$$

$$\hookleftarrow$$

$X_{Node} \in \{1, 2, \ldots, K\}$



$X_A \neq X_B$

$\forall e \in Edges; \ (a,b)$: $X_a \neq X_b$

$X_{Node, colour} \in \{0, 1\}$

$$\sum_i X_{Node, i} = 1$$

$\forall e \in Edge, \ c_1, \in Colour;$

(A, B) $\quad$ $X_{A,c1} \to X_{B,c1}$

$$C_{s,g,l} \to \text{lecturer} \in \{0, 1\}$$

group

start time

$$C_{s,g,l} \to C_s$$

$$X_{c,p,t} \in \{0, 1\}$$

time

course $\quad$ place

$$(X_{c,p,t}) \longrightarrow \sum_{c',p'} X_{c',p',t} = 0$$

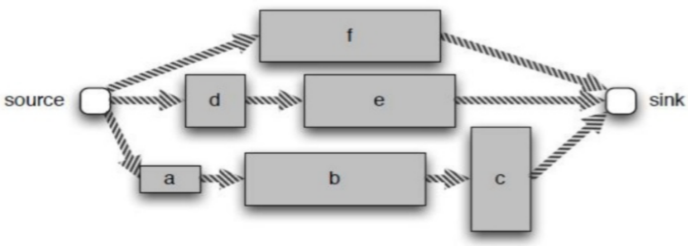
$$c' \in C(c)$$

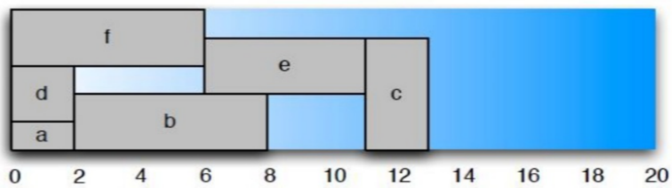
$$X_{c,t} \longrightarrow L_{f(c),t}$$

$$L_{i,t} \rightarrow$$

$$\sum_t L_{i,t} \leq 5$$

Resource-Constrained Project Scheduling Problem





Constraints

Precedence Relations

$task_i < task_j$

Resource Limitation

$task_i = (s_i, D_i, R_i)$

Model the problem using Boolean variables

Image taken "Improving Scheduling by Learning", Andreas Schutt, PhD thesis, University of Melbourne, 2011

$S_{i,t} \in \{0, 1\}$

$\forall i: \sum_t S_{i,t} = 1$

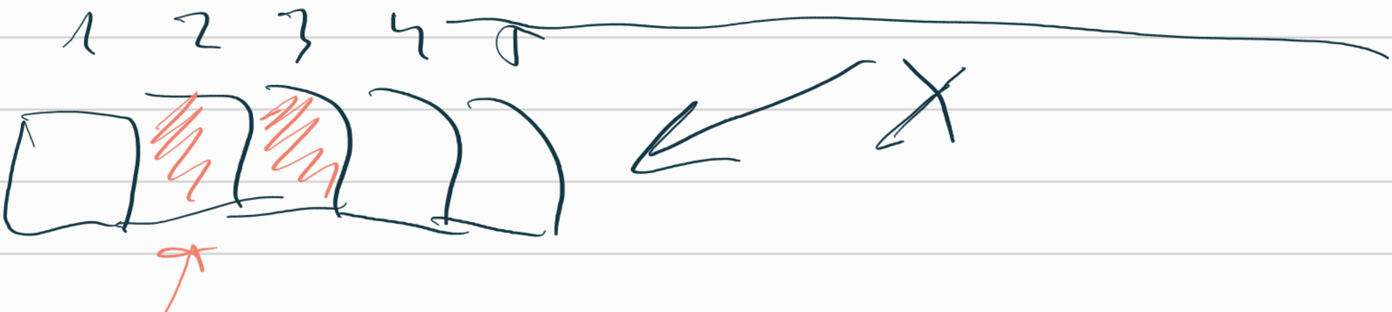
$task_i < task_j$

$$S_{i,t} \rightarrow \sum_{t' \le t+dur(i)-1} S_{j,t'} = 0$$

$$S_{i,t} \rightarrow \sum_{t' \ge t+dur(i)} S_{j,t'} \ge 1$$

$X_{i,t}$

$$\forall t : \sum_i X_{i,t} \cdot R_i \le R_{MAX}$$



$S_{i,2}$ $\forall t, i:$

duration 2

$$S_{i,t} \Leftrightarrow \bigwedge X_{i,t'} \quad t' \in [t, t+1, \ldots, t+Dur_i - 1]$$

Modelling and Search

Emir Demirović

Algorithmics group | TU Delft

Algorithms for NP-Hard Problems (CSE2310 2023)

Lukina, Demirović, Yorke-Smith

Last time…

Modelling

Variables

Constraints

Objective function

"decisions we can make"

"feasibility"

"quality of solution"

Goal for Today

Model-and-Solve Paradigm

Search with Inference/Propagation/Pruning

Problem

$\downarrow$

Model

Problem

concrete

e.g., timetabling

Model

abstract

e.g., equations

unclear

Problem

concrete

e.g., timetabling

unambiguous

Model

abstract

e.g., equations

unclear **Problem** concrete

e.g., timetabling

unambiguous **Model** abstract

e.g., equations

Solve algorithms

Modelling Formalisms

Variables

Constraints

Objective function

Modelling Formalisms

Variables

Constraints

Objective function

Boolean
$x_i \in \{0, 1\}$

Clauses
$(x_i \ \lor \ x_j \ \lor \ \overline{x_z})$

$$\sum_i w_i \cdot x_i$$

Modelling Formalisms

Variables

Boolean

$x_i \in \{0, 1\}$

Constraints

Clauses

$(x_i \ \lor \ x_j \ \lor \ \overline{x_z})$

Objective function

$$\sum_i w_i \cdot x_i$$

(Maximum) Satisfiability

(Max)SAT

Modelling Formalisms

Variables

Constraints

Objective function

Boolean

$x_i \in \{0, 1\}$

Linear Inequalities

$$\sum w_i \cdot x_i \geq k$$

$$\sum_i w_i \cdot x_i$$

Pseudo-Boolean (PB)

Modelling Formalisms

Variables

Integer

$x_i \in \{0, 1, \dots, 10\}$

Constraints

Linear Inequalities

$$\sum w_i \cdot x_i \geq k$$

Objective function

$$\sum_i w_i \cdot x_i$$

Integer Program (IP)

Modelling Formalisms

Variables

Constraints

Objective function

Real-Valued

$x_i \in [0, 30]$

Linear Inequalities

$$\sum w_i \cdot x_i \geq k$$

$$\sum_i w_i \cdot x_i$$

Linear Program (LP)

Modelling Formalisms

Variables	Constraints	Objective function
Integer	Predicates	
$x_i \in \{0, 1, \dots, 10\}$	$C: X^n \rightarrow \{0,1\}$	$\sum_i w_i \cdot x_i$
	$C(x_1, x_2, x_3)$	

Modelling Formalisms

Variables

Constraints

Objective function

Integer

$$x_i \in \{0, 1, \dots, 10\}$$

Predicates

$$C: X^n \rightarrow \{0,1\}$$

$$C(x_1, x_2, x_3)$$

$$\sum_i w_i \cdot x_i$$

Special case: Linear Inequalities

$$\sum w_i \cdot x_i \geq k$$

Modelling Formalisms

Variables	Constraints	Objective function
Integer	Predicates	
$x_i \in \{0, 1, \ldots, 10\}$	$C: X^n \rightarrow \{0,1\}$	$\sum_i w_i \cdot x_i$
	$C(x_1, x_2, x_3)$	

$\boxed{\textbf{Constraint Programming (CP)}}$

Constraint Programming in this course…

Integer Variables

$$x_i \in \{0, 1, 2, \dots, k\}$$

Constraints

$$\sum w_i \cdot x_i \geq k$$

Constraint Programming in this course…

Integer Variables

$$x_i \in \{0, 1, 2, \dots, k\}$$

Constraints

$$\sum w_i \cdot x_i \geq k$$

Special case $x_i \geq x_j$

Constraint Programming in this course…

Integer Variables

$$x_i \in \{0, 1, 2, \dots, k\}$$

Constraints

$$\sum w_i \cdot x_i \geq k$$

Special case $\quad x_i \geq x_j \quad y_i \rightarrow \sum_j w_j \cdot x_j \geq k$

y_i is a Boolean variable

Constraint Programming in this course…

Integer Variables

$$x_i \in \{0, 1, 2, \dots, k\}$$

Constraints

$$\sum w_i \cdot x_i \geq k \qquad x_i \neq x_j$$

Special case $x_i \geq x_j$ $\quad y_i \rightarrow \sum_j w_j \cdot x_j \geq k$

y_i is a Boolean variable

Constraint Programming in this course…

Integer Variables

$$x_i \in \{0, 1, 2, \dots, k\}$$

Constraints

$$\sum w_i \cdot x_i \geq k \qquad x_i \neq x_j$$

More constraints as we go

Special case $\quad x_i \geq x_j \qquad y_i \rightarrow \sum_j w_j \cdot x_j \geq k$

y_i is a Boolean variable

Modelling Formalisms

(Max)SAT

Constraint Programming

Linear Programming

Pseudo-Boolean

Integer Programming

Which formalism do we select?

Modelling Formalisms

(Max)SAT

Constraint Programming

Linear Programming

Pseudo-Boolean

Integer Programming

Which formalism do we select?

Ease of modelling

Underlying algorithmic properties

Modelling Formalisms

MaxSAT/PB/IP/CP are equivalent in theory!

Modelling Formalisms

MaxSAT/PB/IP/CP are equivalent in theory!

Formalisms ties to algorithms

By selecting a formalism, we implicitly select an algorithm

Modelling Formalisms

MaxSAT/PB/IP/CP are equivalent in theory!

Formalisms ties to algorithms

By selecting a formalism, we implicitly select an algorithm

Try out different approaches for a given problem!
Some algorithms better than others for some problems

Modelling Formalisms

Modelling languages like MiniZinc
make the process of evaluating different algorithms easier
(see Lab 3)

Try out different approaches for a given problem!
Some algorithms better than others for some problems

Problem

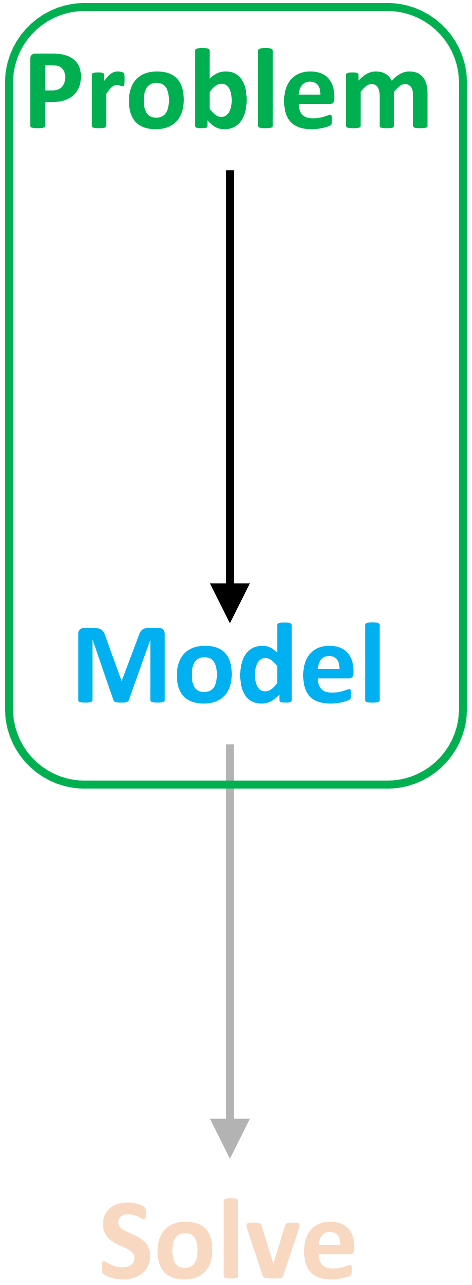
$\downarrow$

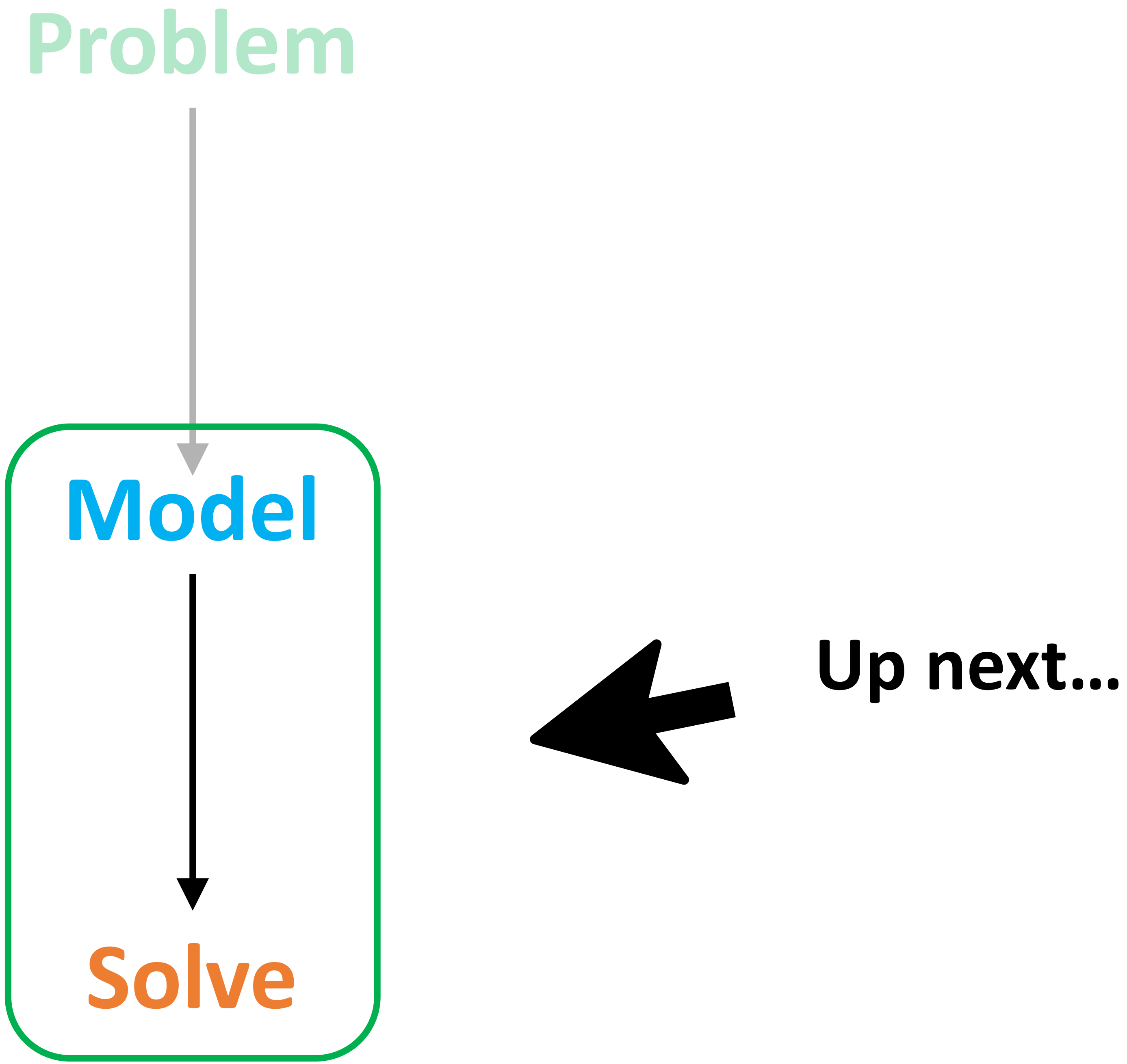
Model

$\downarrow$

Solve

So far...




Problem
Model
Solve
Up next…

Constraint Programming in this course (for now)…

Integer Variables

$$x_i \in \{0, 1, 2, \dots, k\}$$

Constraints

$$\sum w_i \cdot x_i \geq k \qquad\qquad x_i \neq x_j$$

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

"Decision"

ω_1: $x_1 - x_3 \geq 0$

$x_1 = 0 \; @ \; 1$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

$\omega_1\colon\ x_1 - x_3 \geq 0$

$\omega_2\colon\ x_1 + x_2 + x_3 + x_6 \geq 1$

$\omega_3\colon\ x_3 - x_4 - x_5 \geq -1$

$\omega_4\colon\ x_6 - x_2 \geq 0$

$\omega_5\colon\ x_1 - x_2 - x_6 \geq -1$

$\omega_6\colon\ x_4 + x_5 \geq 1$

$\omega_7\colon\ x_4 - x_5 \geq 0$

$\omega_8\colon\ x_5 - x_4 \geq 0$

$x_1 = 0 \text{ @ } 1$

"Decision level"

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

ω_1: $x_1 - x_3 \geq 0$

Search

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

$\boxed{x_1 = 0 \ @\ 1}$

ω_3: $x_3 - x_4 - x_5 \geq -1$

Infer information from constraints which contain x_1

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

Search

$\boxed{x_1 = 0 \ @ \ 1}$

$\omega_1: x_3 = 0 \ @ \ 1$

Constraint $\boldsymbol{\omega_1}$ propagated $\boldsymbol{x_3 = 0}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

$\boxed{x_1 = 0 \ @ \ 1}$

$\omega_1: x_3 = 0 \ @ \ 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

No more propagations possible from constraints linked to x_1

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

$\omega_1: \; x_1 - x_3 \geq 0$

$\omega_2: \; x_1 + x_2 + x_3 + x_6 \geq 1$

$\omega_3: \; x_3 - x_4 - x_5 \geq -1$

$\omega_4: \; x_6 - x_2 \geq 0$

$\omega_5: \; x_1 - x_2 - x_6 \geq -1$

$\omega_6: \; x_4 + x_5 \geq 1$

$\omega_7: \; x_4 - x_5 \geq 0$

$\omega_8: \; x_5 - x_4 \geq 0$

$x_1 = 0 \; @ \; 1$

$\boxed{\omega_1 : x_3 = 0 \; @ \; 1}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0 \; @ \; 1$

$\boxed{\omega_1 : x_3 = 0 \; @ \; 1}$

No propagation possible from constraints linked to x_3

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

No more propagations in this round: "fixed point"

time to make a decision!

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

Decision level 2

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

$\omega_1\colon\ x_1 - x_3 \geq 0$

$\omega_2\colon\ x_1 + x_2 + x_3 + x_6 \geq 1$

$\omega_3\colon\ x_3 - x_4 - x_5 \geq -1$

$\omega_4\colon\ x_6 - x_2 \geq 0$

$\omega_5\colon\ x_1 - x_2 - x_6 \geq -1$

$\omega_6\colon\ x_4 + x_5 \geq 1$

$\omega_7\colon\ x_4 - x_5 \geq 0$

$\omega_8\colon\ x_5 - x_4 \geq 0$

$x_1 = 0 \ @\ 1$

$\omega_1\colon x_3 = 0 \ @\ 1$

$\boxed{x_2 = 0 \ @\ 2}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

$\omega_1\colon\ x_1 - x_3 \geq 0$

$\boxed{\omega_2\colon\ x_1 + x_2 + x_3 + x_6 \geq 1}$

$\omega_3\colon\ x_3 - x_4 - x_5 \geq -1$

$\omega_4\colon\ x_6 - x_2 \geq 0$

$\omega_5\colon\ x_1 - x_2 - x_6 \geq -1$

$\omega_6\colon\ x_4 + x_5 \geq 1$

$\omega_7\colon\ x_4 - x_5 \geq 0$

$\omega_8\colon\ x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

$\omega_1\colon x_3 = 0$ @ 1

$\boxed{x_2 = 0 \text{ @ } 2}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

$\boxed{\omega_2\text{: } x_1 + x_2 + x_3 + x_6 \geq 1}$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$\boxed{x_2 = 0 \text{ @ } 2}$

ω_2: $x_6 = 1$ @ 2

Constraint ω_2 propagates x_6

Taken from exam prep April 2021

Search

$x_i \in \{0, 1\}$

$\omega_1\colon\ x_1 - x_3 \geq 0$

$\omega_2\colon\ x_1 + x_2 + x_3 + x_6 \geq 1$

$\omega_3\colon\ x_3 - x_4 - x_5 \geq -1$

$\boxed{\omega_4\colon\ x_6 - x_2 \geq 0}$

$\omega_5\colon\ x_1 - x_2 - x_6 \geq -1$

$\omega_6\colon\ x_4 + x_5 \geq 1$

$\omega_7\colon\ x_4 - x_5 \geq 0$

$\omega_8\colon\ x_5 - x_4 \geq 0$

$x_1 = 0\ @\ 1$

$\omega_1\colon x_3 = 0\ @\ 1$

$\boxed{x_2 = 0\ @\ 2}$

$\omega_2\colon x_6 = 1\ @\ 2$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

$\omega_1\colon\ x_1 - x_3 \geq 0$

$\omega_2\colon\ x_1 + x_2 + x_3 + x_6 \geq 1$

$\omega_3\colon\ x_3 - x_4 - x_5 \geq -1$

$\boxed{\omega_4\colon\ x_6 - x_2 \geq 0}$

$\omega_5\colon\ x_1 - x_2 - x_6 \geq -1$

$\omega_6\colon\ x_4 + x_5 \geq 1$

$\omega_7\colon\ x_4 - x_5 \geq 0$

$\omega_8\colon\ x_5 - x_4 \geq 0$

$x_1 = 0 \ @\ 1$

$\omega_1\colon x_3 = 0 \ @\ 1$

$\boxed{x_2 = 0 \ @\ 2}$

$\omega_2\colon x_6 = 1 \ @\ 2$

Constraint ω_4 satisfied, no propagation

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

$\boxed{\omega_5: \; x_1 - x_2 - x_6 \geq -1}$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$\boxed{x_2 = 0 \text{ @ } 2}$

ω_2: $x_6 = 1$ @ 2

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

$\boxed{\omega_5\text{: } x_1 - x_2 - x_6 \geq -1}$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$\boxed{x_2 = 0 \text{ @ } 2}$

ω_2: $x_6 = 1$ @ 2

Satisfied constraint, no propagation

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

$\boxed{\omega_2: x_6 = 1 \text{ @ } 2}$

Consider constraints linked to x_6, no propagation possible

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

At a fixed point, time to make a decision!

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

$\boxed{x_4 = 0 \text{ @ } 3}$

Taken from exam prep April 2021

Search

$x_i \in \{0,1\}$

$\omega_1\colon\ x_1 - x_3 \geq 0$

$\omega_2\colon\ x_1 + x_2 + x_3 + x_6 \geq 1$

$\boxed{\omega_3\colon\ x_3 - x_4 - x_5 \geq -1}$

$\omega_4\colon\ x_6 - x_2 \geq 0$

$\omega_5\colon\ x_1 - x_2 - x_6 \geq -1$

$\omega_6\colon\ x_4 + x_5 \geq 1$

$\omega_7\colon\ x_4 - x_5 \geq 0$

$\omega_8\colon\ x_5 - x_4 \geq 0$

no propagation

$x_1 = 0\ @\ 1$

$\omega_1\colon x_3 = 0\ @\ 1$

$x_2 = 0\ @\ 2$

$\omega_2\colon x_6 = 1\ @\ 2$

$\boxed{x_4 = 0\ @\ 3}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

$\boxed{\omega_6\text{: } x_4 + x_5 \geq 1}$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

$\boxed{x_4 = 0 \text{ @ } 3}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

$\boxed{\omega_6\text{: } x_4 + x_5 \geq 1}$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

$\boxed{x_4 = 0 \text{ @ } 3}$

ω_3: $x_5 = 1$ @ 3

Propagate $x_5 = 1$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

$\boxed{\omega_7\text{: } x_4 - x_5 \geq 0}$

ω_8: $x_5 - x_4 \geq 0$

$\mathbf{-1 \geq 0}$

“Conflict”

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

$\boxed{x_4 = 0 \text{ @ } 3}$

ω_3: $x_5 = 1$ @ 3

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

$\boxed{\omega_7\text{: } x_4 - x_5 \geq 0}$

ω_8: $x_5 - x_4 \geq 0$

$\mathbf{-1 \geq 0}$

"Conflict"

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

$\boxed{x_4 = 0 \text{ @ } 3}$

ω_3: $x_5 = 1$ @ 3

Current partial assignment infeasible, backtrack from last decision level!

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

$\omega_1 : x_3 = 0$ @ 1

$x_2 = 0$ @ 2

$\omega_2 : x_6 = 1$ @ 2

$\boxed{x_4 \cancel{=} 0 \ @ \ 3}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

$x_4 \neq 0$ @ 2

$x_4 = 1$ @ 2

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

$x_4 \neq 0$ @ 2

$\boxed{x_4 = 1 \text{ @ } 2}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

$\boxed{\omega_3: \ x_3 - x_4 - x_5 \geq -1}$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 = 0$ @ 2

ω_2: $x_6 = 1$ @ 2

$x_4 \neq 0$ @ 2

$\boxed{x_4 = 1}$ @ 2

ω_3: $x_5 = 0$ @ 2

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

$\boxed{\omega_8\text{: } x_5 - x_4 \geq 0}$

$x_1 = 0 \ @\ 1$

ω_1: $x_3 = 0 \ @\ 1$

$x_2 = 0 \ @\ 2$

ω_2: $x_6 = 1 \ @\ 2$

$x_4 \neq 0 \ @\ 2$

$\boxed{x_4 = 1 \ @\ 2}$

ω_3: $x_5 = 0 \ @\ 2$

$\mathbf{0 \geq 1}$

"Conflict"

Taken from exam prep April 2021

Search

$x_i \in \{0, 1\}$

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

$\omega_1: x_3 = 0$ @ 1

$\boxed{x_2 \cancel{=} 0 \text{ @ } 2}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

ω_5: $x_1 - x_2 - x_6 \geq -1$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0$ @ 1

ω_1: $x_3 = 0$ @ 1

$x_2 \neq 0$ @ 1

$\boxed{x_2 = 1 \text{ @ } 1}$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

$\omega_1\colon\ x_1 - x_3 \geq 0$

$\omega_2\colon\ x_1 + x_2 + x_3 + x_6 \geq 1$

$\omega_3\colon\ x_3 - x_4 - x_5 \geq -1$

$\boxed{\omega_4\colon\ x_6 - x_2 \geq 0}$

$\omega_5\colon\ x_1 - x_2 - x_6 \geq -1$

$\omega_6\colon\ x_4 + x_5 \geq 1$

$\omega_7\colon\ x_4 - x_5 \geq 0$

$\omega_8\colon\ x_5 - x_4 \geq 0$

Propagate $x_6 = 1$

$x_1 = 0 \ @\ 1$

$\omega_1\colon x_3 = 0 \ @\ 1$

$x_2 \neq 0 \ @\ 1$

$\boxed{x_2 = 1 \ @\ 1}$

$\omega_4\colon x_6 = 1 \ @\ 1$

Taken from exam prep April 2021

$x_i \in \{0, 1\}$

Search

ω_1: $x_1 - x_3 \geq 0$

ω_2: $x_1 + x_2 + x_3 + x_6 \geq 1$

ω_3: $x_3 - x_4 - x_5 \geq -1$

ω_4: $x_6 - x_2 \geq 0$

$\boxed{\omega_5: \; x_1 - x_2 - x_6 \geq -1}$

ω_6: $x_4 + x_5 \geq 1$

ω_7: $x_4 - x_5 \geq 0$

ω_8: $x_5 - x_4 \geq 0$

$x_1 = 0 \; @ \; 1$

$\omega_1: x_3 = 0 \; @ \; 1$

$x_2 \neq 0 \; @ \; 1$

$\boxed{x_2 = 1 \; @ \; 1}$

$\omega_4: x_6 = 1 \; @ \; 1$

$\mathbf{0 \geq 1}$

"Conflict"

Taken from exam prep April 2021

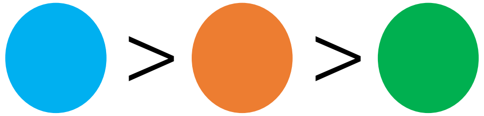
General Strategy

Make a decision

Check constraints for propagation until fixed point

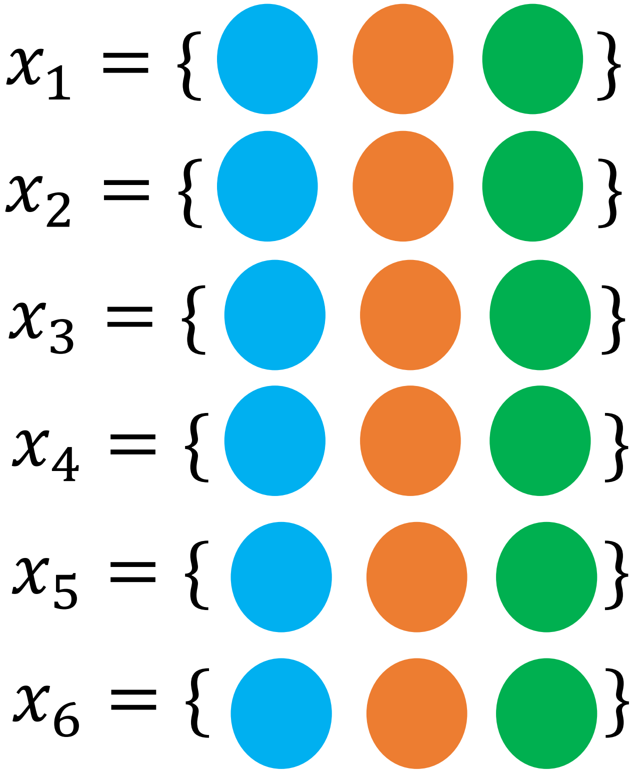
Backtrack if conflict

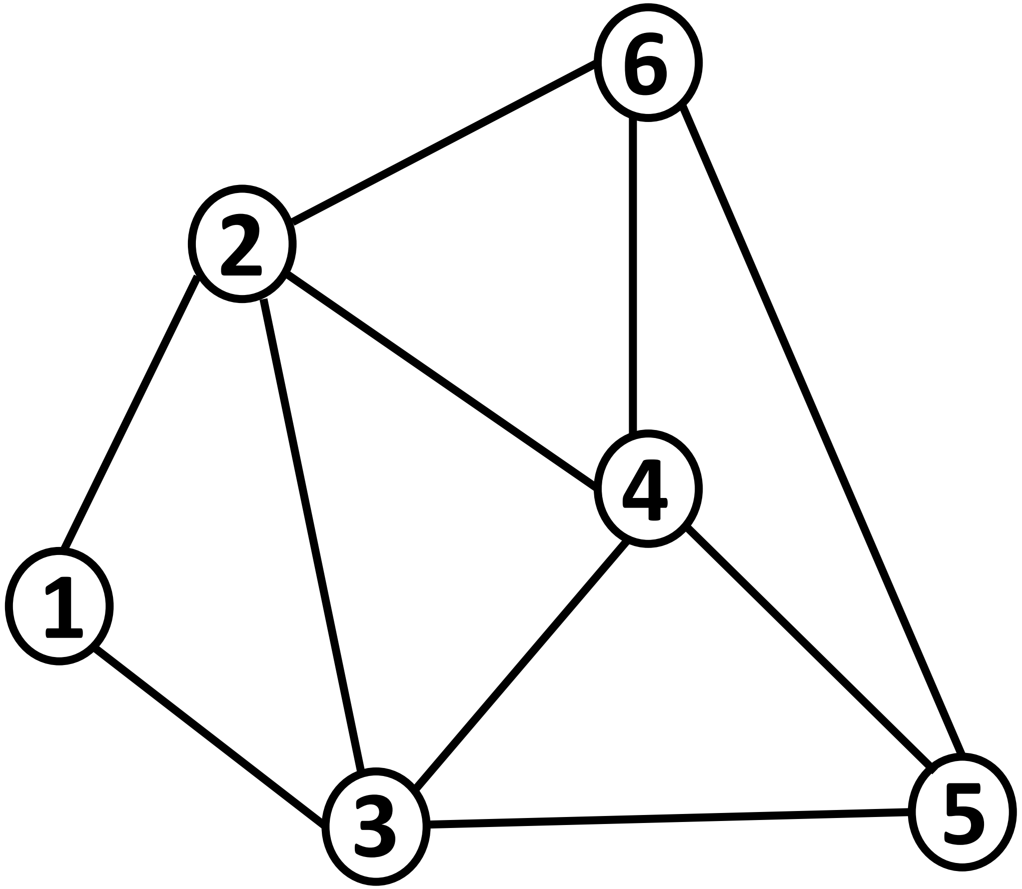
Graph Colouring



$$x_1 > x_2 > \cdots > x_6$$

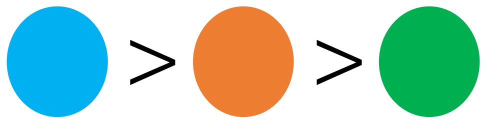
$$x_i \neq x_j$$





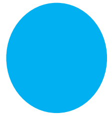
Graph Colouring

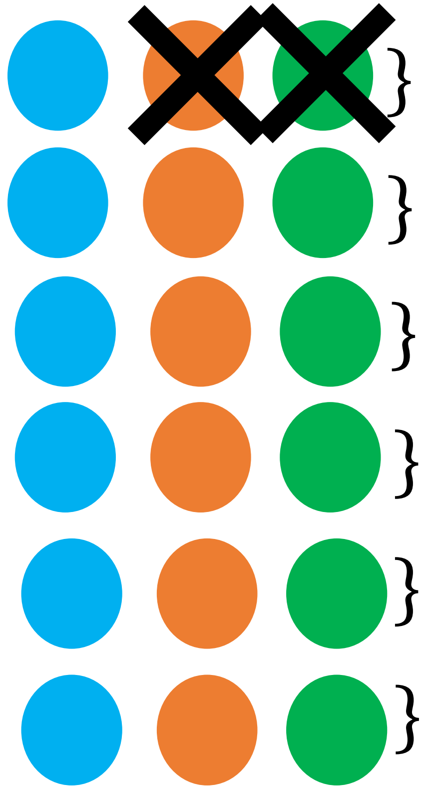
$x_i \neq x_j$

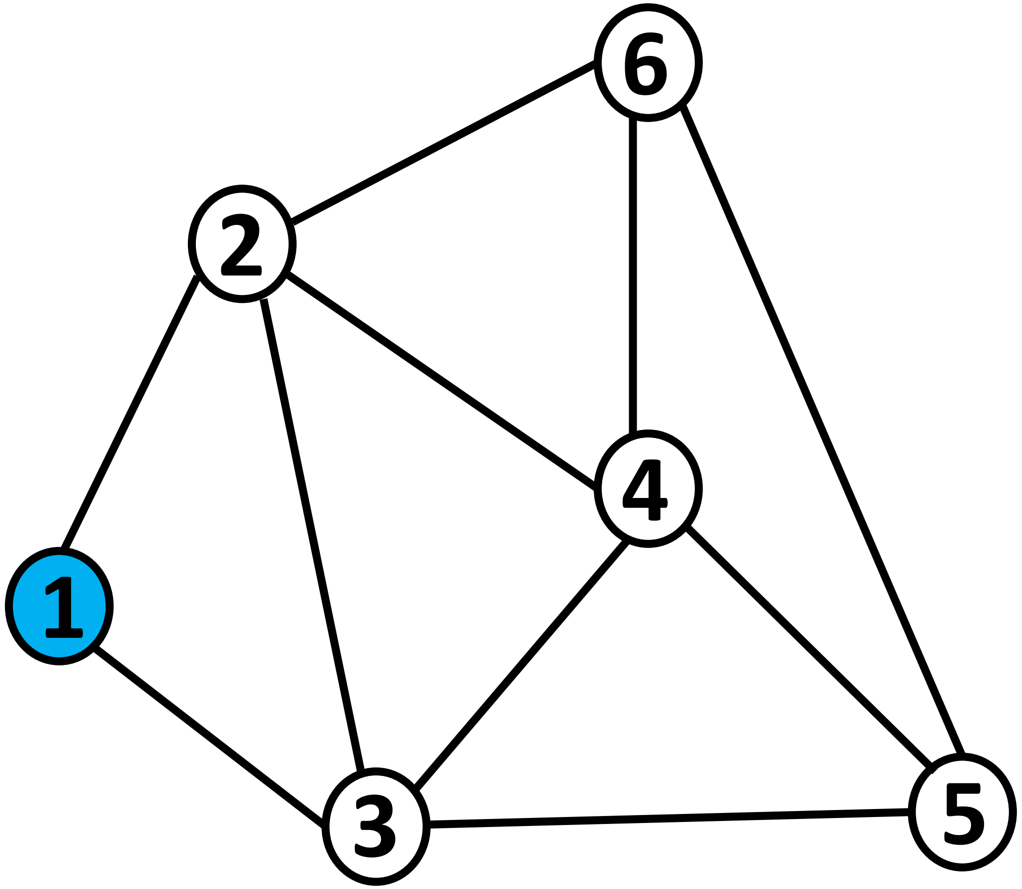


$x_1 > x_2 > \cdots > x_6$

decision

$x_1 =$  @ 1





Graph Colouring

$x_i \neq x_j$

$x_1 > x_2 > \cdots > x_6$

$x_1 =$ @ 1

$x_2 \neq$ @ 1

$x_3 \neq$ @ 1

propagations

$x_1 = \{\ \}$

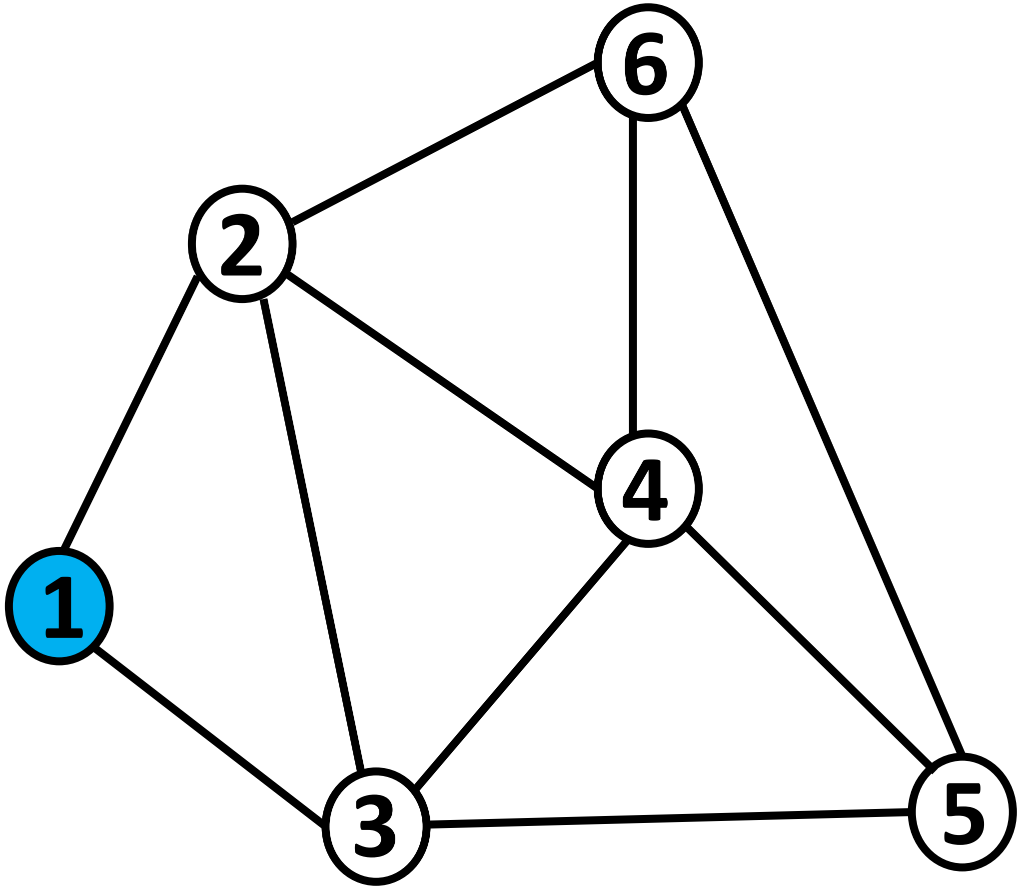
$x_2 = \{\ \}$

$x_3 = \{\ \}$

$x_4 = \{\ \}$

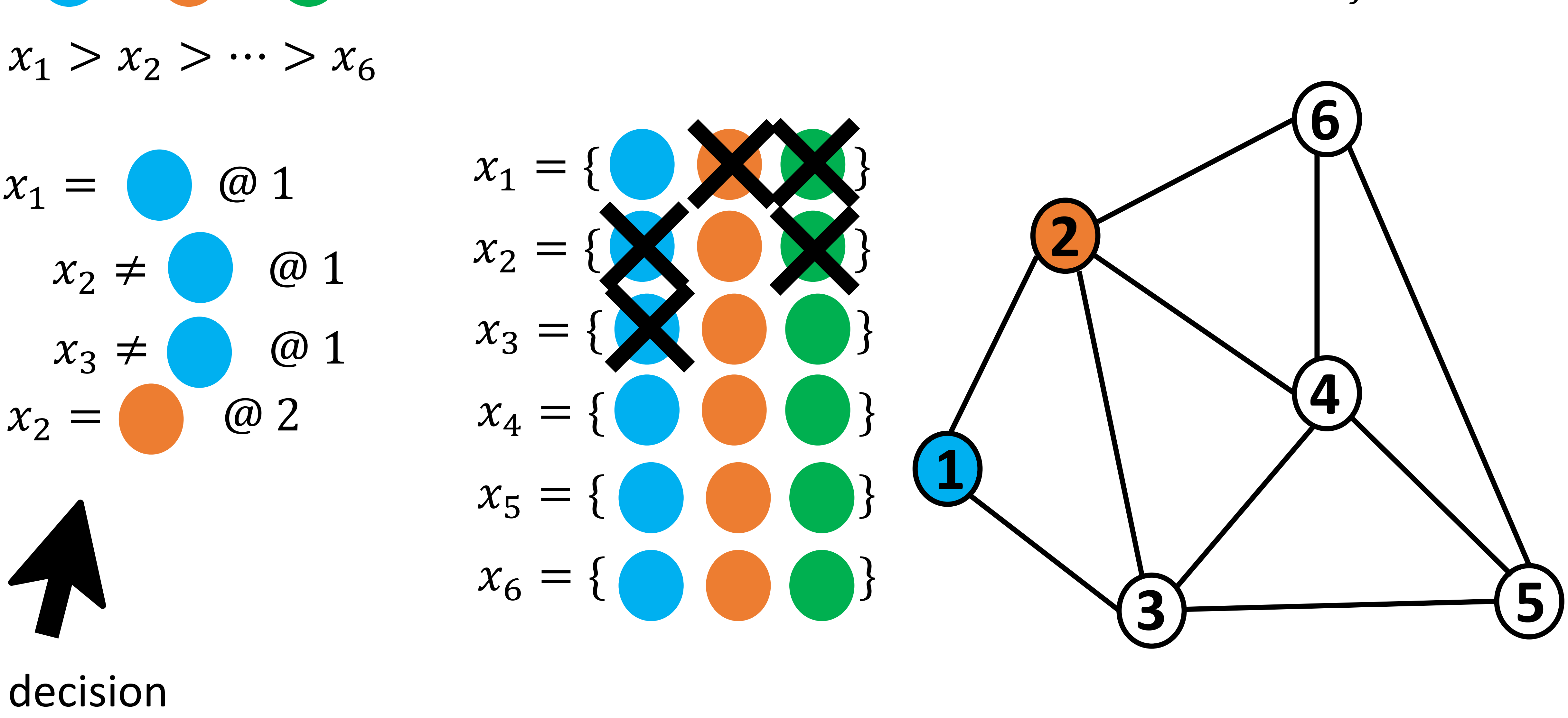
$x_5 = \{\ \}$

$x_6 = \{\ \}$

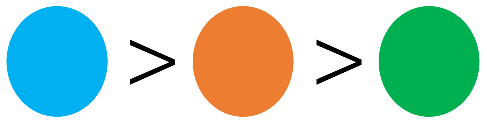


Taken from resit exam June 2021

Graph Colouring



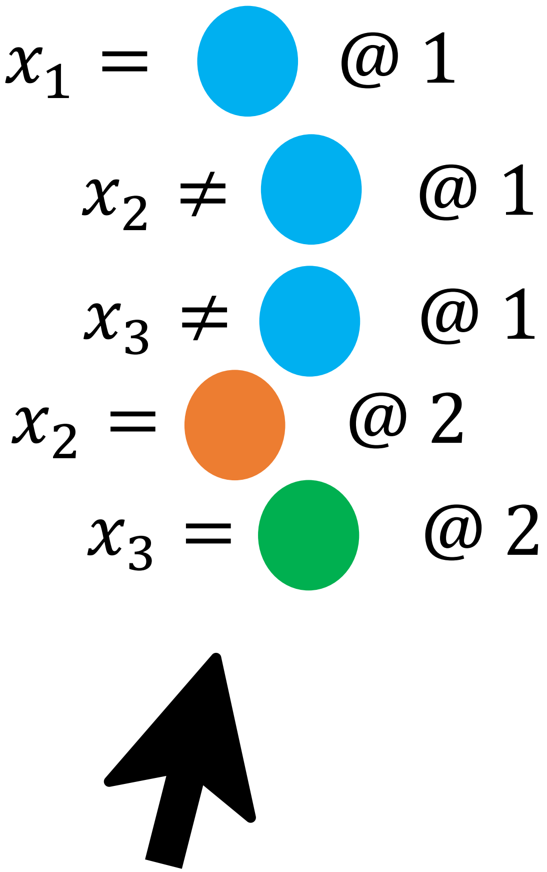
Taken from resit exam June 2021

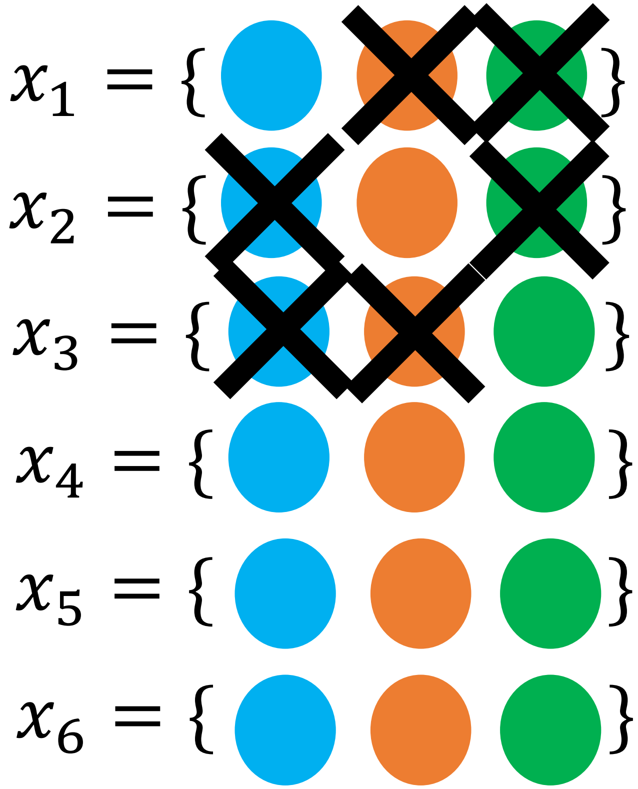


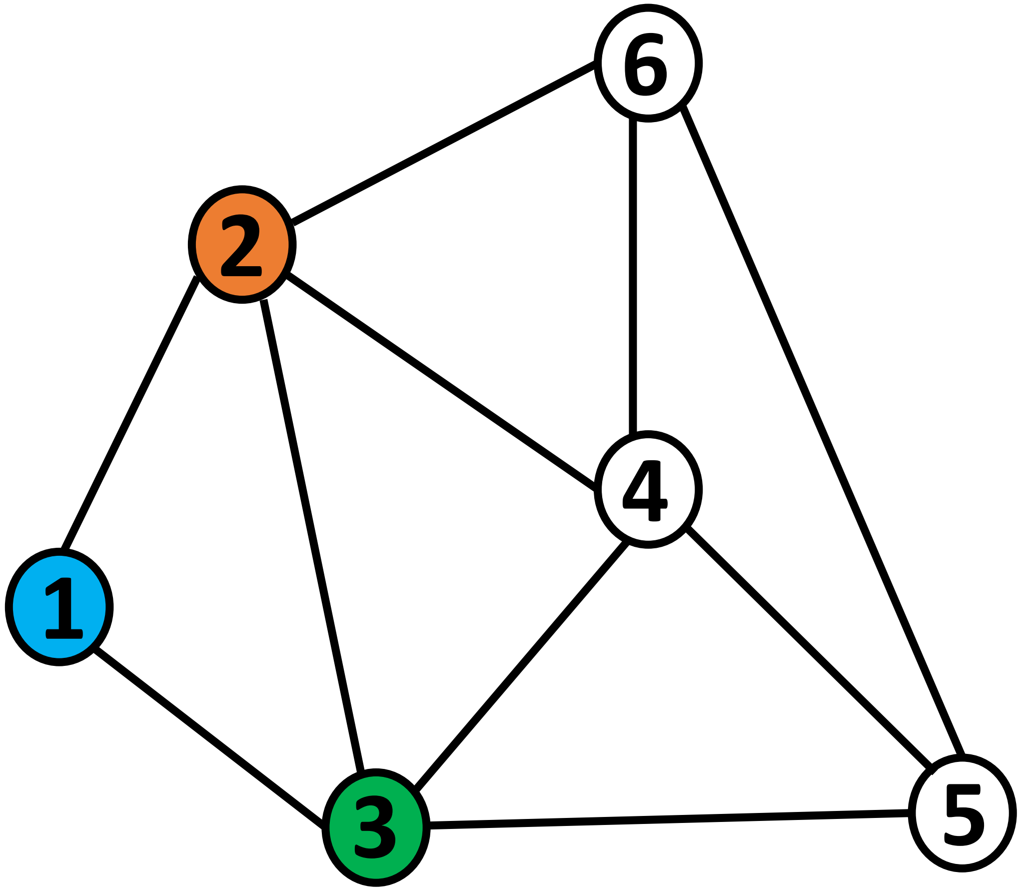
Graph Colouring

$x_i \neq x_j$

$x_1 > x_2 > \cdots > x_6$





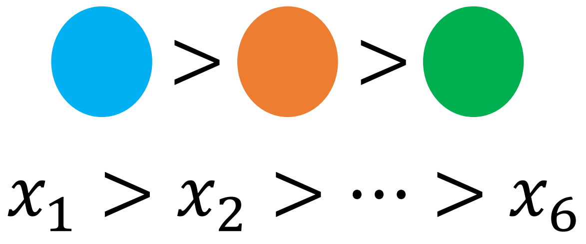


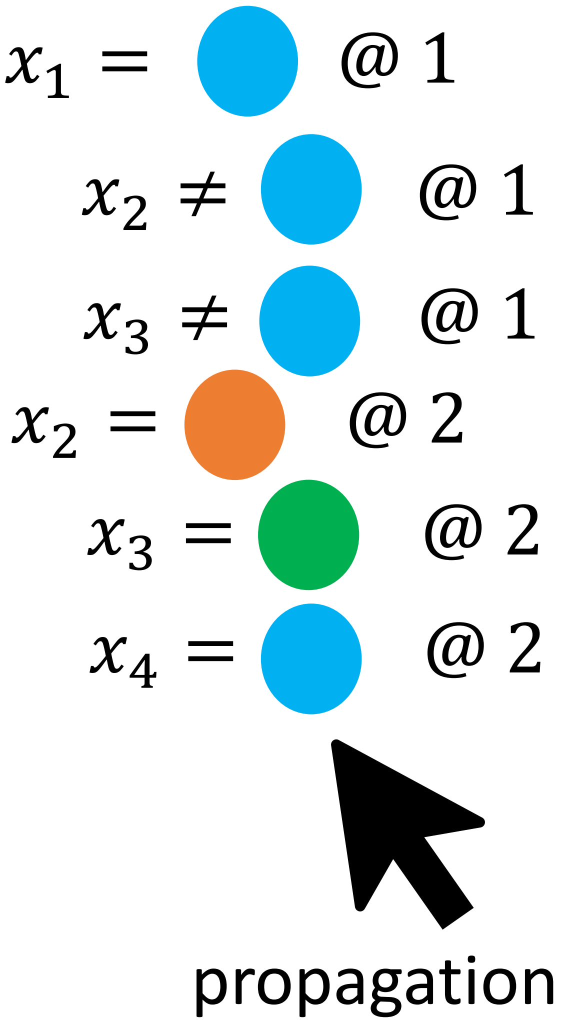
propagation

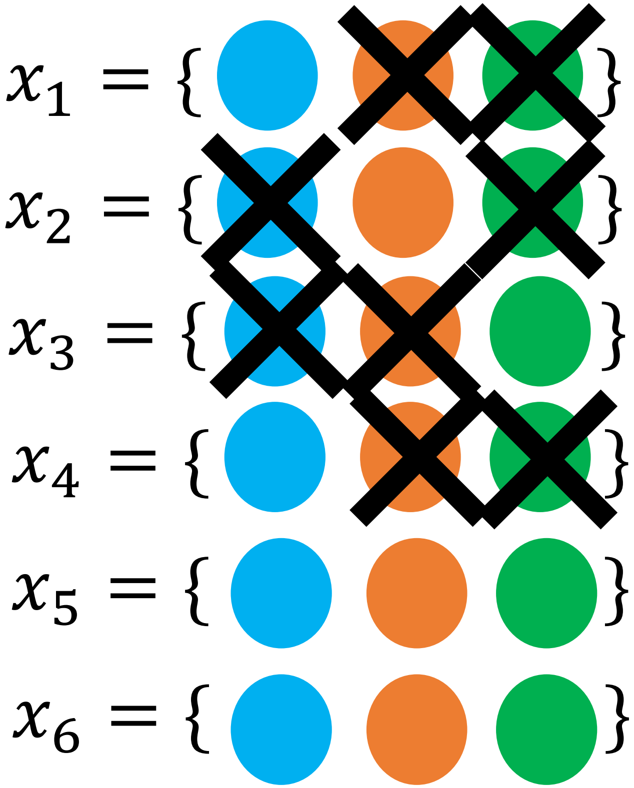
Taken from resit exam June 2021

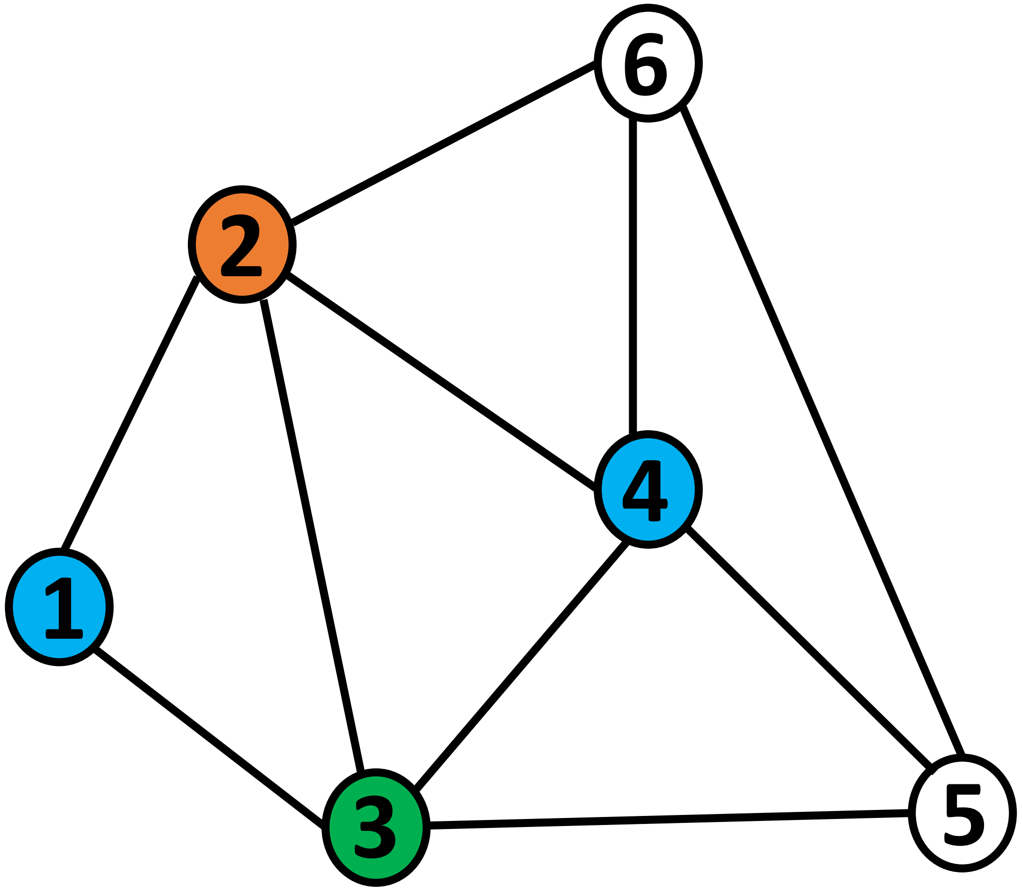
Graph Colouring

$x_i \neq x_j$

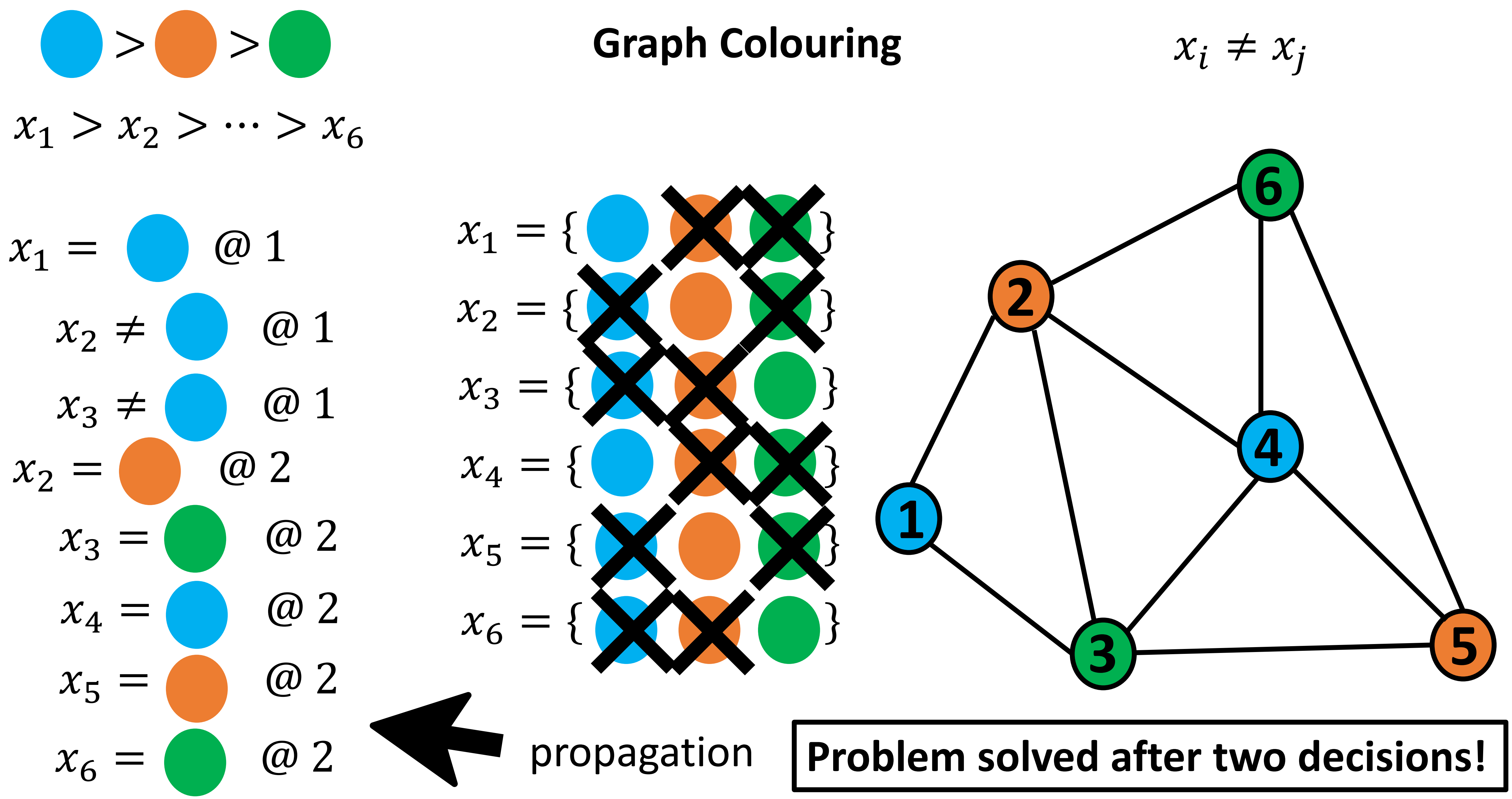








Taken from resit exam June 2021



Taken from resit exam June 2021

Search
(with Pruning/Inference/Propagation)

1. Select a variable

(with at least two values in the domain)

2. Assign a value from the domain of the variable

3. Propagate

(until fixed point or conflict)

4. Conflict?

(a variable has empty domain, constraint violated)

Search
(with Pruning/Inference/Propagation)

1. Select a variable
(with at least two values in the domain)

2. Assign a value from the domain of the variable

3. Propagate
(until fixed point or conflict)

"make a decision"

4. Conflict?
(a variable has empty domain, constraint violated)

Search
(with Pruning/Inference/Propagation)

1. Select a variable

(with at least two values in the domain)

2. Assign a value from the domain of the variable

3. Propagate

(until fixed point or conflict)

4. Conflict?

(a variable has empty domain, constraint violated)

Variable Selection Heuristic

“Which unassigned variable to select next?”

What is a good strategy?

Variable Selection Heuristic

"Which unassigned variable to select next?"

What is a good strategy?

We want to…

…find a solution with few decisions

…reach a conflict as soon as possible

Many possible strategies to choose from,
highly depends on the problem!

Variable Selection Heuristic

"Which unassigned variable to select next?"

Random → not great, ok as baseline

Variable Selection Heuristic

"Which unassigned variable to select next?"

Random → not great, ok as baseline

...with smallest domain

"first fail"

Variable Selection Heuristic

"Which unassigned variable to select next?"

Random → not great, ok as baseline

...with smallest domain

"first fail"

...with smallest value in its domain

Variable Selection Heuristic

"Which unassigned variable to select next?"

Random → not great, ok as baseline

...with smallest domain
"first fail"

...with smallest value in its domain

Select variable that was often part of a conflicting constraint

Variable Selection Heuristic

"Which unassigned variable to select next?"

Random → not great, ok as baseline

…with smallest domain

"first fail"

…with smallest value in its domain

Select variable that was often part of a conflicting constraint

Custom variable ordering tailored to the problem

Value Selection Heuristic

"Given an unassigned variable, decide on the next value from its domain"

What is a good strategy?

Value Selection Heuristic

"Given an unassigned variable, decide on the next value from its domain"

What is a good strategy?

We want to…

…select value that is part of a feasible solution

…reach a conflict as soon as possible

Many possible strategies to choose from,
highly depends on the problem!

Value Selection Heuristic

"Given an unassigned variable, decide on the next value from its domain"

Random → not great, ok as baseline

Value Selection Heuristic

"Given an unassigned variable, decide on the next value from its domain"

Random → not great, ok as baseline

Smallest/largest value first

"schedule tasks as early as possible"

Value Selection Heuristic

"Given an unassigned variable, decide on the next value from its domain"

Random → not great, ok as baseline

Smallest/largest value first

According to a reference (infeasible) solution
e.g., obtain via a greedy approach

Value Selection Heuristic

"Given an unassigned variable, decide on the next value from its domain"

Random → not great, ok as baseline

Smallest/largest value first

According to a reference (infeasible) solution
e.g., obtain via a greedy approach

Split the domain
$x_i \geq k$

Search
(with Pruning/Inference/Propagation)

1. Select a variable
(with at least two values in the domain)

2. Assign a value from the domain of the variable

3. Propagate
(until fixed point or conflict)

“make a decision”

4. Conflict?
(a variable has empty domain, constraint violated)

See: https://www.minizinc.org/doc-2.5.5/en/mzn_search.html

Search
(with Pruning/Inference/Propagation)

1. Select a variable

(with at least two values in the domain)

2. Assign a value from the domain of the variable

3. Propagate

(until fixed point or conflict)

Constraints!

4. Conflict?

(a variable has empty domain, constraint violated)

Propagation

"Which values can be removed from domains?"

Each constraint has a propagation algorithm

Propagation

"Which values can be removed from domains?"

Each constraint has a propagation algorithm

$$x \neq y$$

$x \in D_1$ $\qquad\qquad$ $y \in D_2$

When will this constraint propagate?

Propagation

"Which values can be removed from domains?"

Each constraint has a propagation algorithm

$$x \neq y$$

$x \in D_1$ $\quad$ $y \in D_2$

$\boldsymbol{D_i = \{a\}}$ $\rightarrow$ **remove 'a' from** $\boldsymbol{D_j}$

Propagation

"Which values can be removed from domains?"

Each constraint has a propagation algorithm

$$x \neq y$$

$x \in D_1$ $y \in D_2$

$\boldsymbol{D_i = \{a\}}$ $\rightarrow$ **remove 'a' from** $\boldsymbol{D_j}$

$\boldsymbol{x = \{1, 2\}}$

$\boldsymbol{y = \{2, 3, 4, 5\}}$

No propagation

Propagation

"Which values can be removed from domains?"

Each constraint has a propagation algorithm

$$x \neq y$$

$x \in D_1$ $\quad$ $y \in D_2$

$\boldsymbol{D_i = \{a\}}$ $\rightarrow$ **remove 'a' from** $\boldsymbol{D_j}$

$x = \{1, 2\}$	$x = \{2\}$
$y = \{2, 3, 4, 5\}$	$y = \{2, 3, 4, 5\}$
No propagation	Propagate $y \neq 2$

Propagation

"Which values can be removed from domains?"

$$x \geq y$$

$x \in D_1$ $\quad\quad$ $y \in D_2$

What is the propagation algorithm?

Propagation

"Which values can be removed from domains?"

$$x \geq y$$

$x \in D_1$ $\quad$ $y \in D_2$

$$\boldsymbol{x = \{1, 2, 3\}}$$
$$\boldsymbol{y = \{1, 2, 3, 4, 5\}}$$

Propagation

"Which values can be removed from domains?"

$$x \geq y$$

$x \in D_1$ $y \in D_2$

$$\boldsymbol{x = \{1, 2, 3\}}$$
$$\boldsymbol{y = \{1, 2, 3, \cancel{4}, \cancel{5}\}}$$

Propagation

"Which values can be removed from domains?"

$$\boxed{\begin{array}{c} x \geq y \\ x \in D_1 \qquad\qquad y \in D_2 \end{array}}$$

$$\boldsymbol{x = \{1, 2, 3\}}$$
$$\boldsymbol{y = \{1, 2, 3, \cancel{4}, \cancel{5}\}}$$

$$\boxed{\boldsymbol{y \leq UpperBound(x)}}$$

Propagation

"Which values can be removed from domains?"

$$x \geq y$$

$x \in D_1$ $y \in D_2$

$x = \{1, 2, 3\}$
$y = \{1, 2, 3, \cancel{4}, \cancel{5}\}$

$\boldsymbol{x = \{1, 2, 3, 4, 5\}}$
$\boldsymbol{y = \{4, 5\}}$

Propagation

"Which values can be removed from domains?"

$$x \geq y$$

$x \in D_1$ $y \in D_2$

$x = \{1, 2, 3\}$
$y = \{1, 2, 3, \cancel{4}, \cancel{5}\}$

$x = \{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5\}$
$y = \{4, 5\}$

Propagation

"Which values can be removed from domains?"

$$x \geq y$$

$x \in D_1$ $\qquad$ $y \in D_2$

$$\boldsymbol{x = \{1, 2, 3\}}$$
$$\boldsymbol{y = \{1, 2, 3, \cancel{4, 5}\}}$$

$$\boldsymbol{x = \{\cancel{1, 2, 3}, 4, 5\}}$$
$$\boldsymbol{y = \{4, 5\}}$$

$$\boxed{\boldsymbol{x \geq LowerBound(y)}}$$

Propagation

"Which values can be removed from domains?"

$$\boxed{\begin{array}{ccc} & x \geq y & \\ x \in D_1 & & y \in D_2 \end{array}}$$

$x = \{1, 2, 3\}$

$y = \{1, 2, 3, \cancel{4}, \cancel{5}\}$

$x = \{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5\}$

$y = \{4, 5\}$

Dynamically keep track during search

$$\boxed{\boldsymbol{y \leq UpperBound(x)}}$$

$$\boxed{\boldsymbol{x \geq LowerBound(y)}}$$

Propagation

"Which values can be removed from domains?"

$$x \geq y$$

$x \in D_1$ $\qquad$ $y \in D_2$

$$\boldsymbol{x = \{1, 2, 3\}}$$
$$\boldsymbol{y = \{1, 2, 3, 4, 5\}}$$

Propagation

"Which values can be removed from domains?"

$$x \geq y$$

$x \in D_1$ $\quad$ $y \in D_2$

$$\boldsymbol{x = \{1, 2, 3\}}$$
$$\boldsymbol{y = \{1, 2, 3, \cancel{4}, \cancel{5}\}}$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

“Upper Bound”

$$y_2 = UB(y_2) = 2$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

$$y_2 = UB(y_2) = 2$$
$$y_3 = UB(y_3) = ?$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

$$y_2 = UB(y_2) = 2$$
$$y_3 = UB(y_3) = 4$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3,4\}$$

$$y_2 = UB(y_2) = 2$$
$$y_3 = UB(y_3) = 4$$
$$LB(y_1) = ?$$

"Lower Bound"

Linear Inequality Propagator

$$\boxed{y_1 + 2y_2 + 3y_3 \geq 17}$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

"optimistic assignment"

$$\boxed{\begin{array}{l} y_2 = UB(y_2) = 2 \\ y_3 = UB(y_3) = 4 \end{array}}$$

$$LB(y_1) = ?$$

Linear Inequality Propagator

$$\boxed{y_1 + 2y_2 + 3y_3 \geq 17}$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

$$y_2 = UB(y_2) = 2$$
$$y_3 = UB(y_3) = 4$$
$$LB(y_1) = ?$$

$$\boxed{y_1 + 2 \cdot 2 + 3 \cdot 4 \geq 17}$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3,4\}$$

$$y_2 = UB(y_2) = 2$$
$$y_3 = UB(y_3) = 4$$
$$LB(y_1) = ?$$

$$y_1 + 2 \cdot 2 + 3 \cdot 4 \geq 17$$

$$\boxed{y_1 + 16 \geq 17}$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

$$y_2 = UB(y_2) = 2$$
$$y_3 = UB(y_3) = 4$$
$$LB(y_1) = 1$$

$$y_1 + 2 \cdot 2 + 3 \cdot 4 \geq 17$$

$$\boxed{y_1 + 16 \geq 17}$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{\cancel{0}, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

Value ‘0’ removed by propagation

$$y_2 = UB(y_2) = 2$$
$$y_3 = UB(y_3) = 4$$
$$\boxed{LB(y_1) = 1}$$

$$y_1 + 2 \cdot 2 + 3 \cdot 4 \geq 17$$

$$y_1 + 16 \geq 17$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

$$y_1 = UB(y_2) = 4$$
$$y_3 = UB(y_3) = 4$$
$$LB(y_2) = ?$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{1, 4\}$$

$$y_2 \in \{\cancel{0}, 2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

Value ‘0’ removed by propagation

$$y_1 = UB(y_2) = 4$$
$$y_3 = UB(y_3) = 4$$
$$LB(y_2) = 2$$

$$4 + 2 \cdot y_2 + 3 \cdot 4 \geq 17$$

$$y_2 + 16 \geq 17$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{1, 4\}$$

$$y_2 \in \{2\}$$

$$y_3 \in \{0, 1, 2, 3, 4\}$$

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

$$y_1 \in \{1, 4\}$$

$$y_2 \in \{2\}$$

$$y_3 \in \{\cancel{0}, \cancel{1}, \cancel{2}, 3, 4\}$$

Values '0', '1', '2' removed by propagation

Linear Inequality Propagator

$$y_1 + 2y_2 + 3y_3 \geq 17$$

After propagation

$$y_1 \in \{1, 4\}$$

$$y_2 \in \{2\}$$

$$y_3 \in \{3,4\}$$

Before propagation

$$y_1 \in \{0, 1, 4\}$$

$$y_2 \in \{0, 2\}$$

$$y_3 \in \{0, 1, 2, 3,4\}$$

Linear Inequality Propagator

$$\sum w_i \cdot y_i \geq k$$

Propagator

$$y_i \geq \left\lceil \frac{k - \sum_{j \neq i} w_j \cdot UB(y_j)}{w_i} \right\rceil$$

(assuming $w_i \geq 0$)

Linear Inequality Propagator

$$\sum w_i \cdot y_i \geq k$$

set other variables
to the optimistic assignment

Propagator

$$y_i \geq \left\lceil \frac{k - \sum_{j \neq i} w_j \cdot \boxed{UB(y_j)}}{w_i} \right\rceil$$

(assuming $w_i \geq 0$)

Linear Inequality Propagator

$$\sum w_i \cdot y_i \geq k$$

units that
must be covered by
the term $w_i \cdot y_i$

Propagator

$$y_i \geq \left\lceil \frac{\boxed{k - \sum_{j \neq i} w_j \cdot UB(y_j)}}{w_i} \right\rceil$$

(assuming $w_i \geq 0$)

Search

Which part is the most expensive?

1. Select a variable

(with at least two values in the domain)

2. Assign a value from the domain of the variable

3. Propagate

(until fixed point or conflict)

4. Conflict?

(a variable has empty domain, constraint violated)

Search

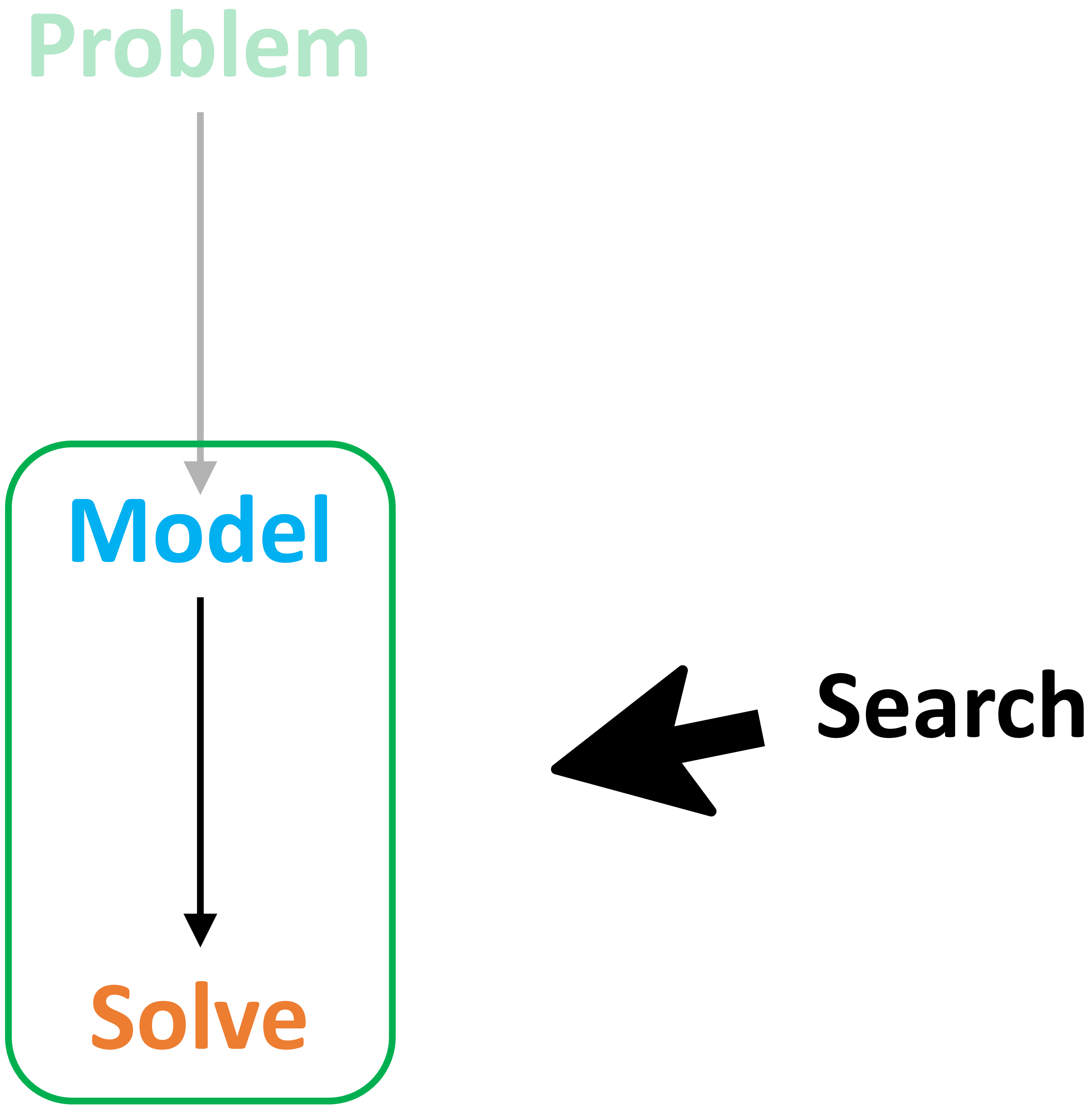
1. Select a variable
(with at least two values in the domain)

2. Assign a value from the domain of the variable

Often
80%+ time
spent here

3. Propagate
(until fixed point or conflict)

4. Conflict?
(a variable has empty domain, constraint violated)


Problem
Model
Solve
Search

Summary

Model-and-Solve Paradigm

Modelling

Constraint Programming

Search

Variable Selection

Value Selection

Propagation

Modelling and Search, Part 2

Emir Demirović

Algorithmics group | TU Delft

Algorithms for NP-Hard Problems (CSE2310 2023)

Lukina, Demirović, Yorke-Smith

Last time…

Model-and-Solve Paradigm

Modelling

Constraint Programming

Search

Variable Selection

Value Selection

Propagation

Problem

Variables

Constraints

Model

Objective function

Solve

Goal for today…

Search Tree

Exponential Growth

Modelling Patterns

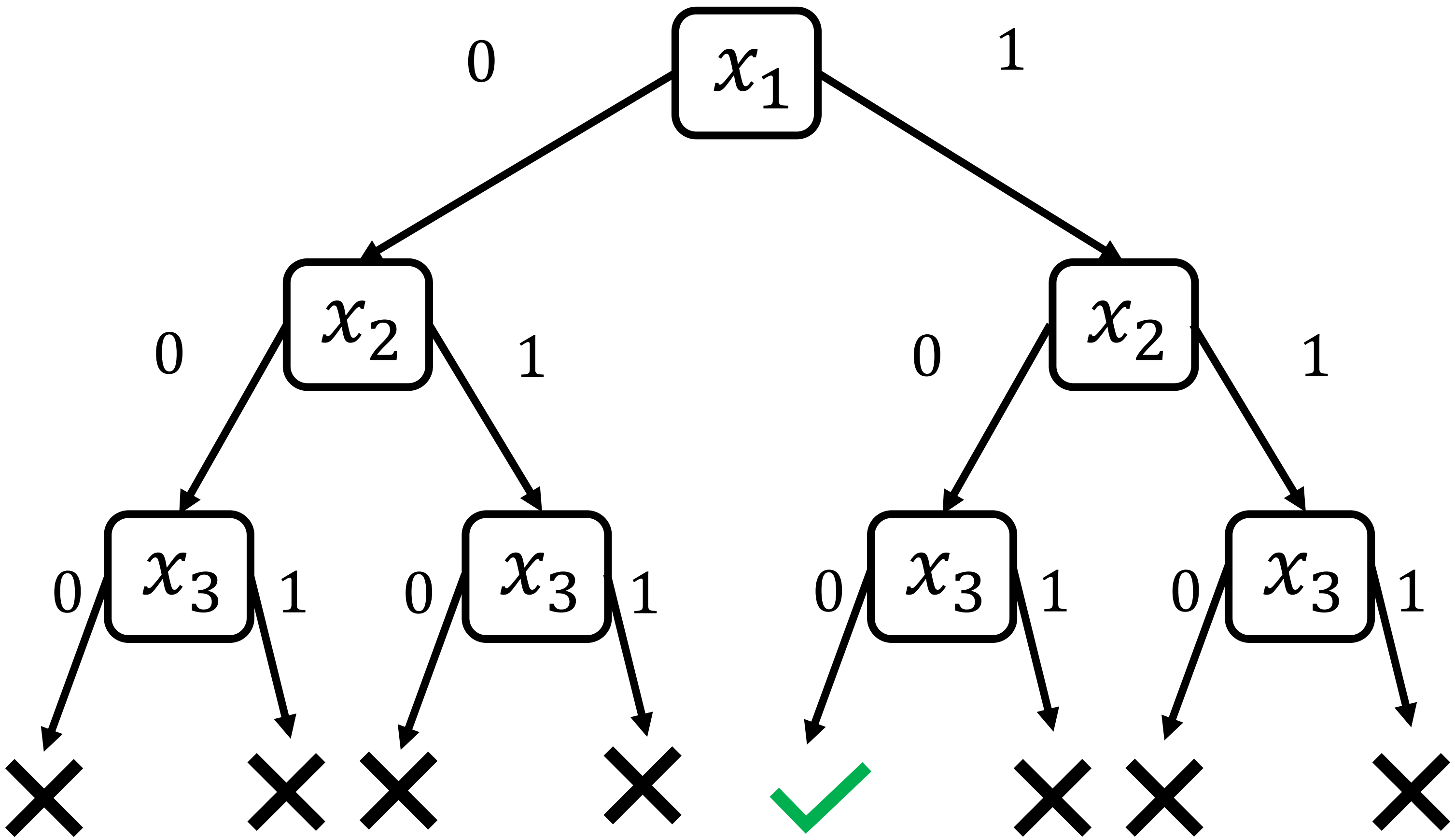
Global Constraints

Symmetry breaking

Search can be represented as a tree

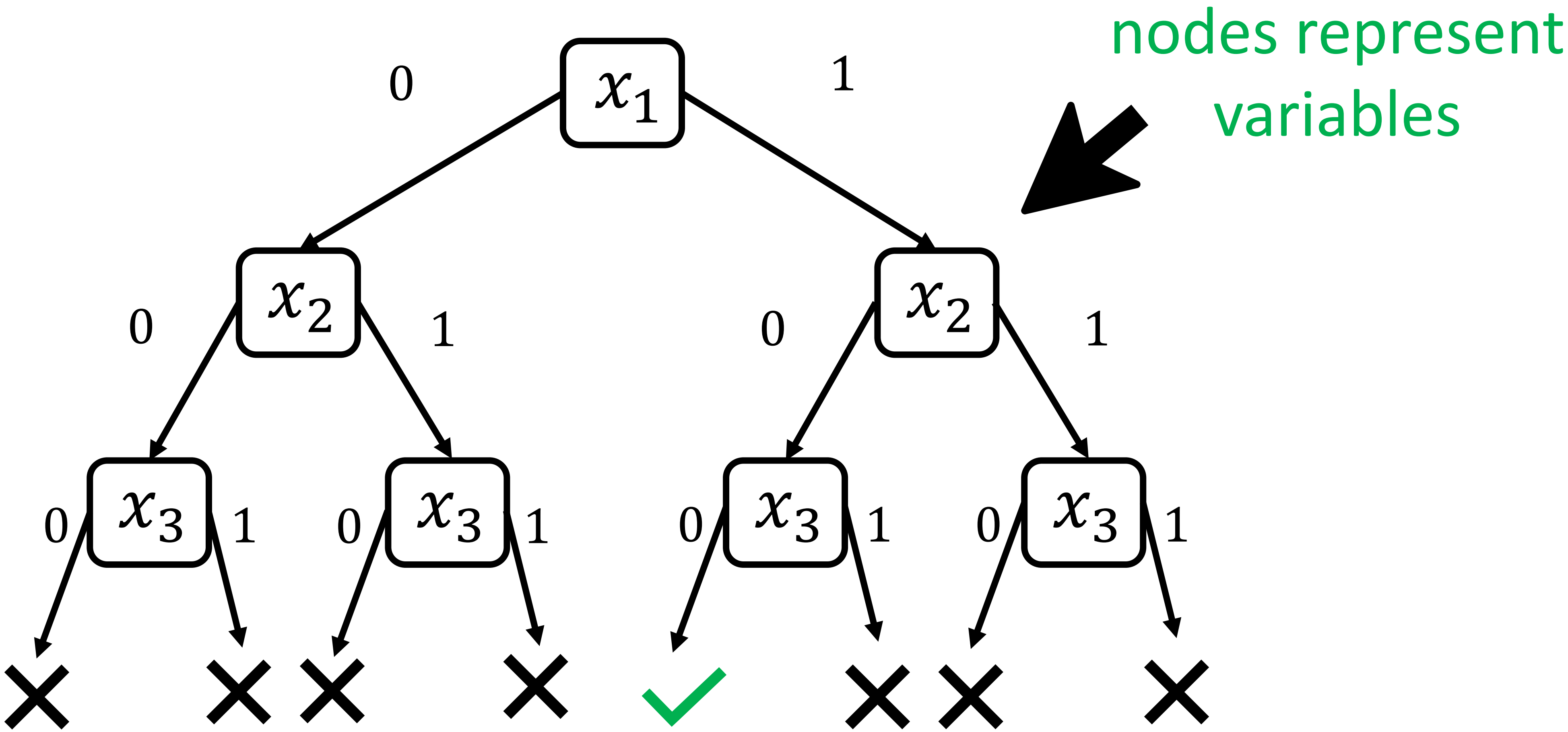
Search can be represented as a tree

“enumerate all feasible solutions”



Search can be represented as a tree

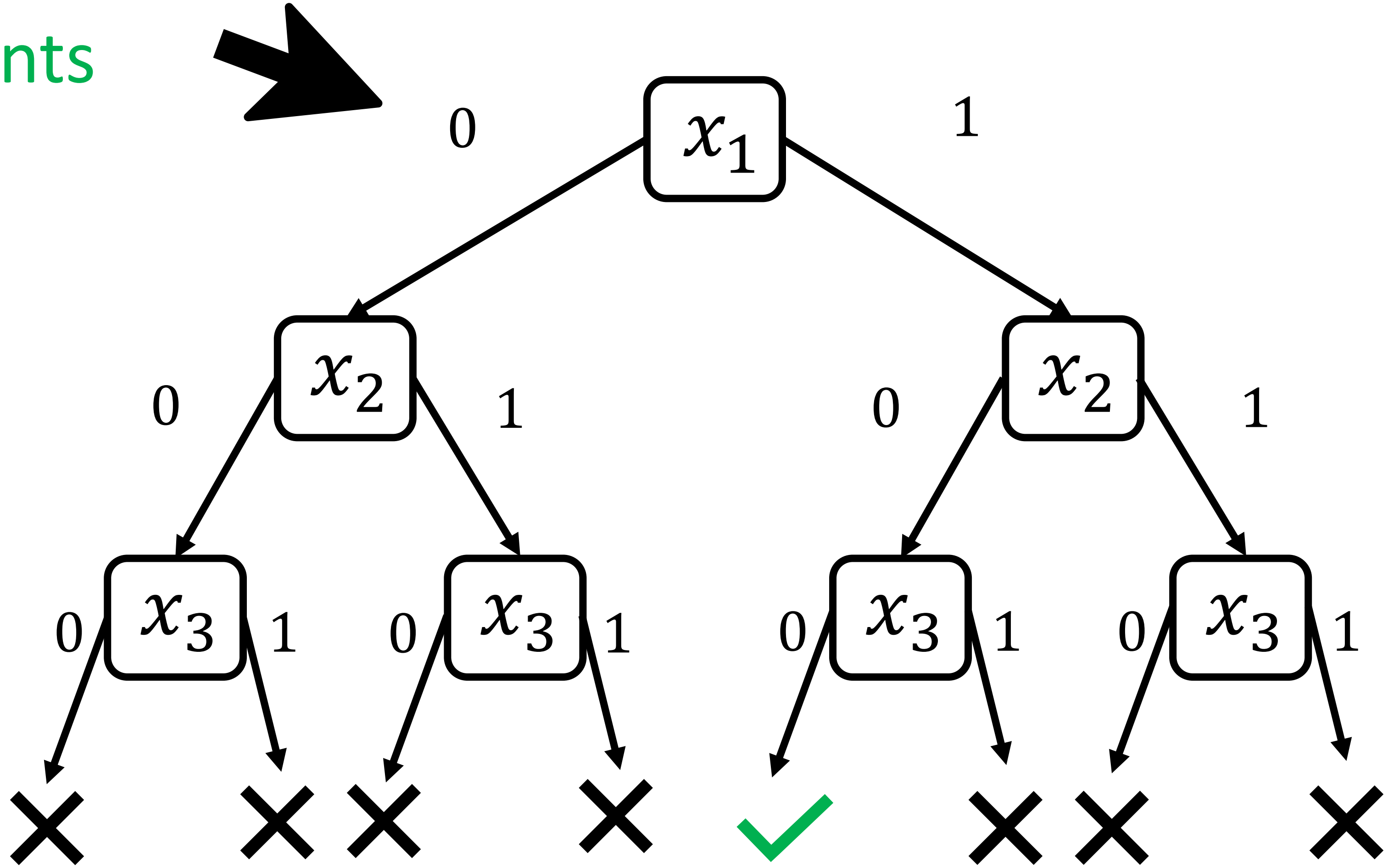
"enumerate all feasible solutions"



Search can be represented as a tree

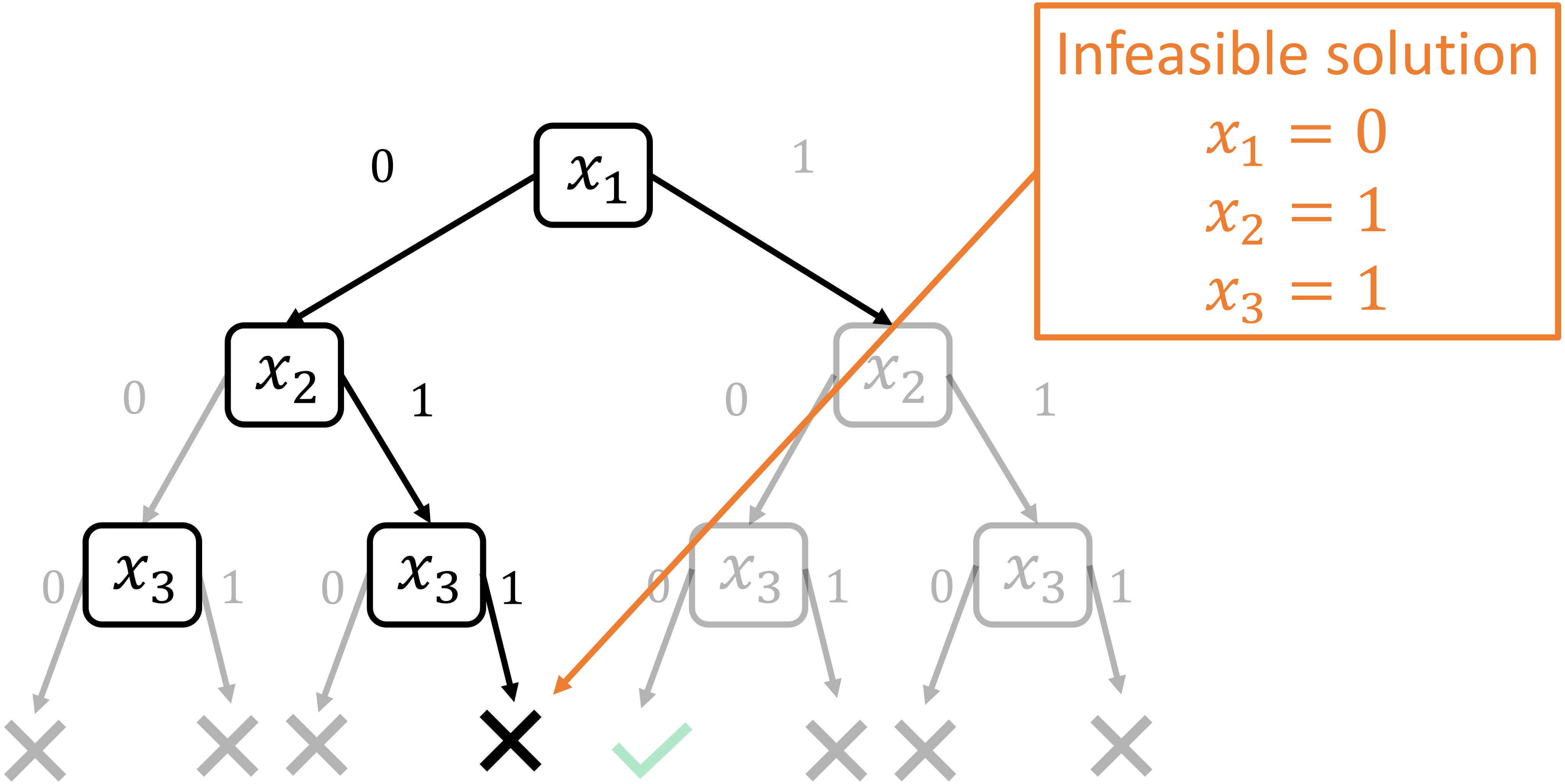
“enumerate all feasible solutions”

edges represent assignments



Search can be represented as a tree

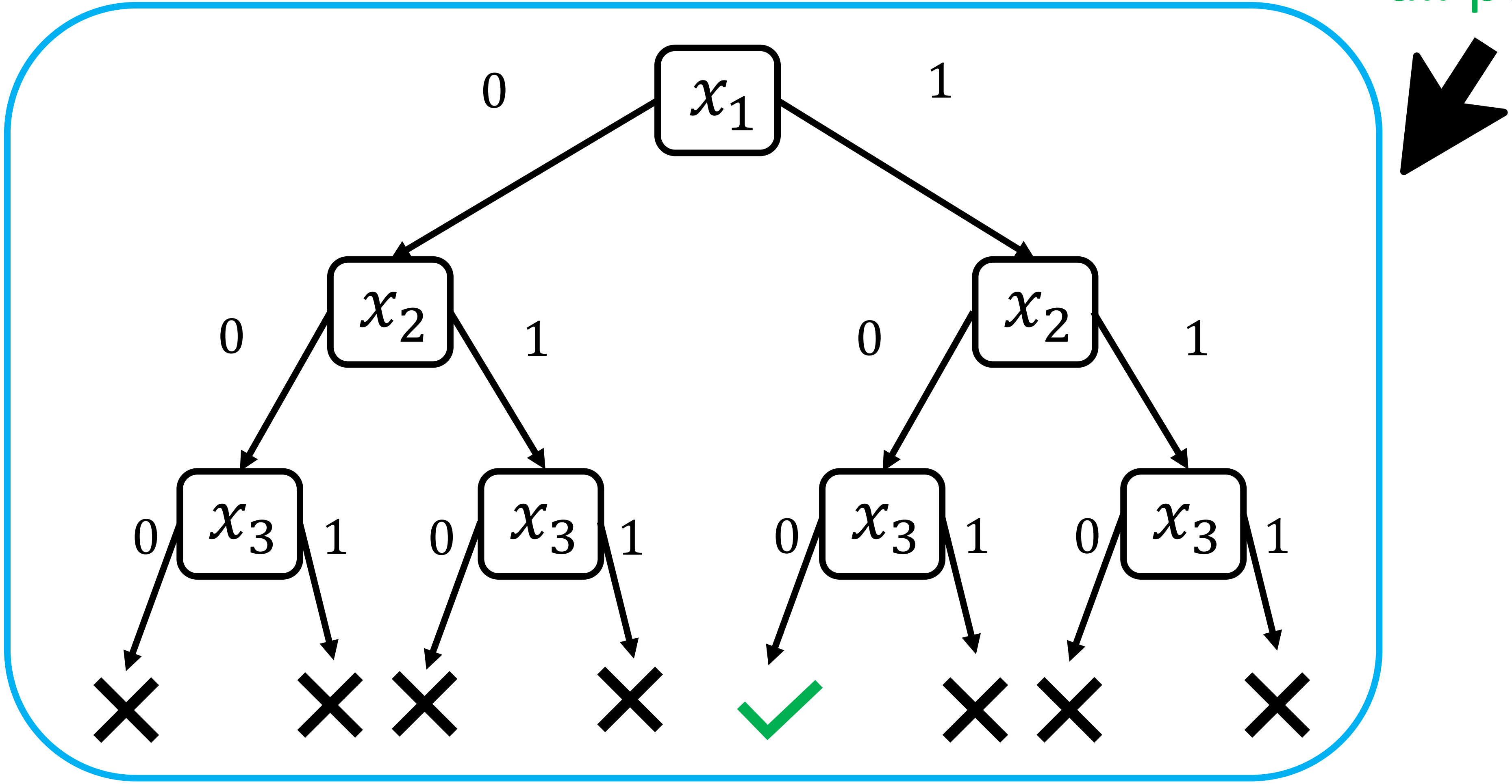
"enumerate all feasible solutions"



Search can be represented as a tree

"enumerate all feasible solutions"

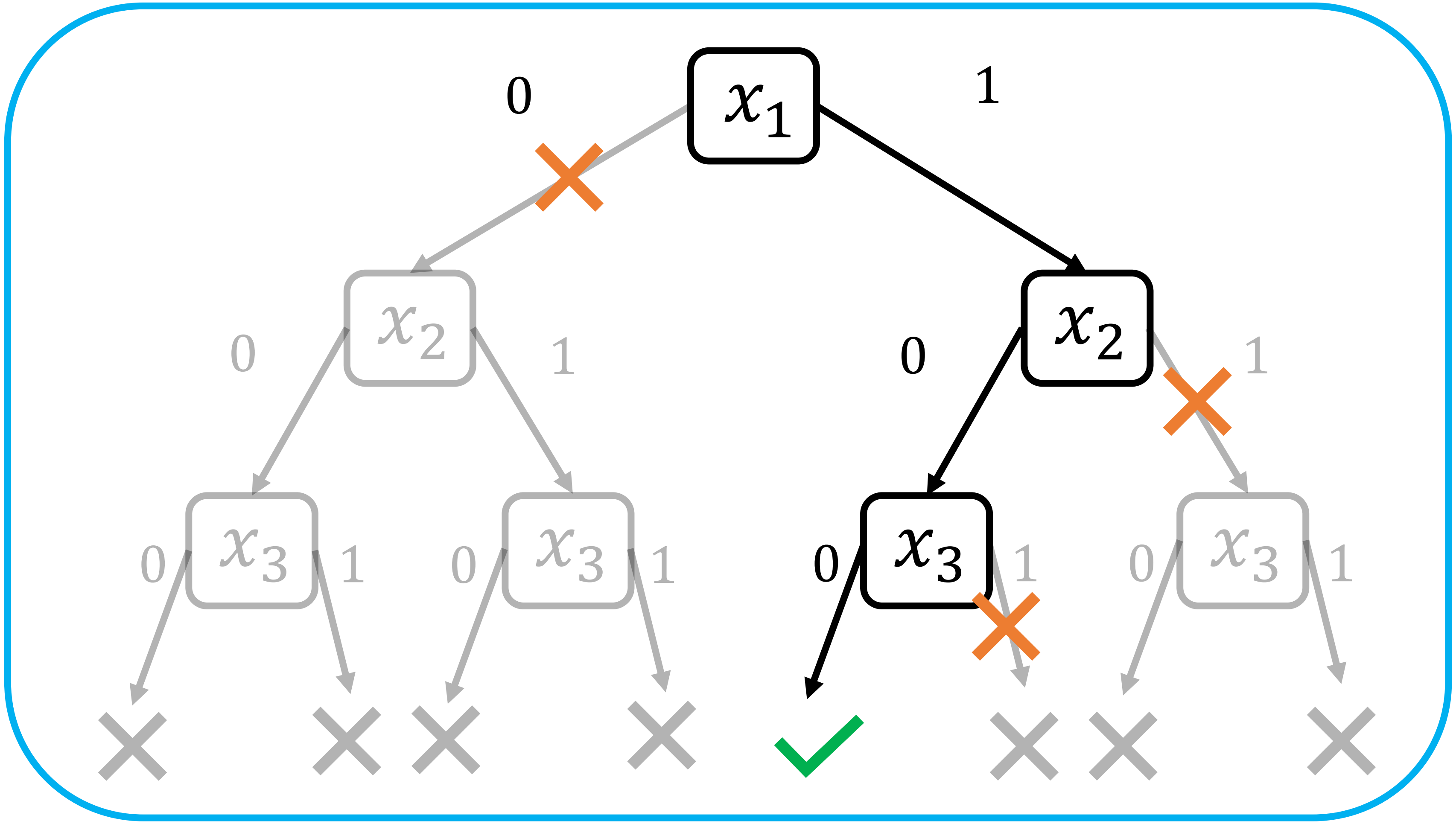
Brute force always includes all paths



Propagation removes edges!

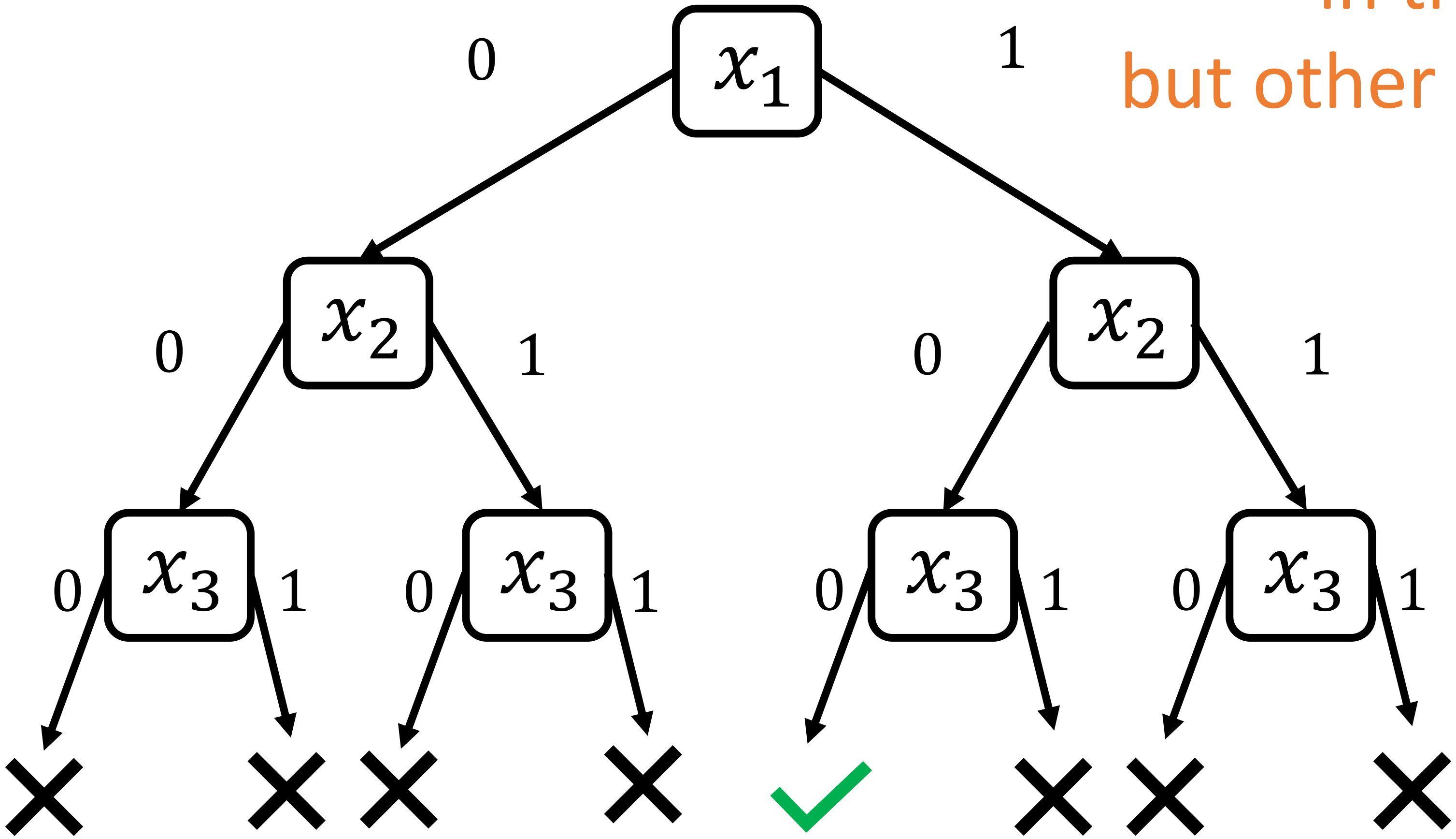
Search can be represented as a tree

"enumerate all feasible solutions"

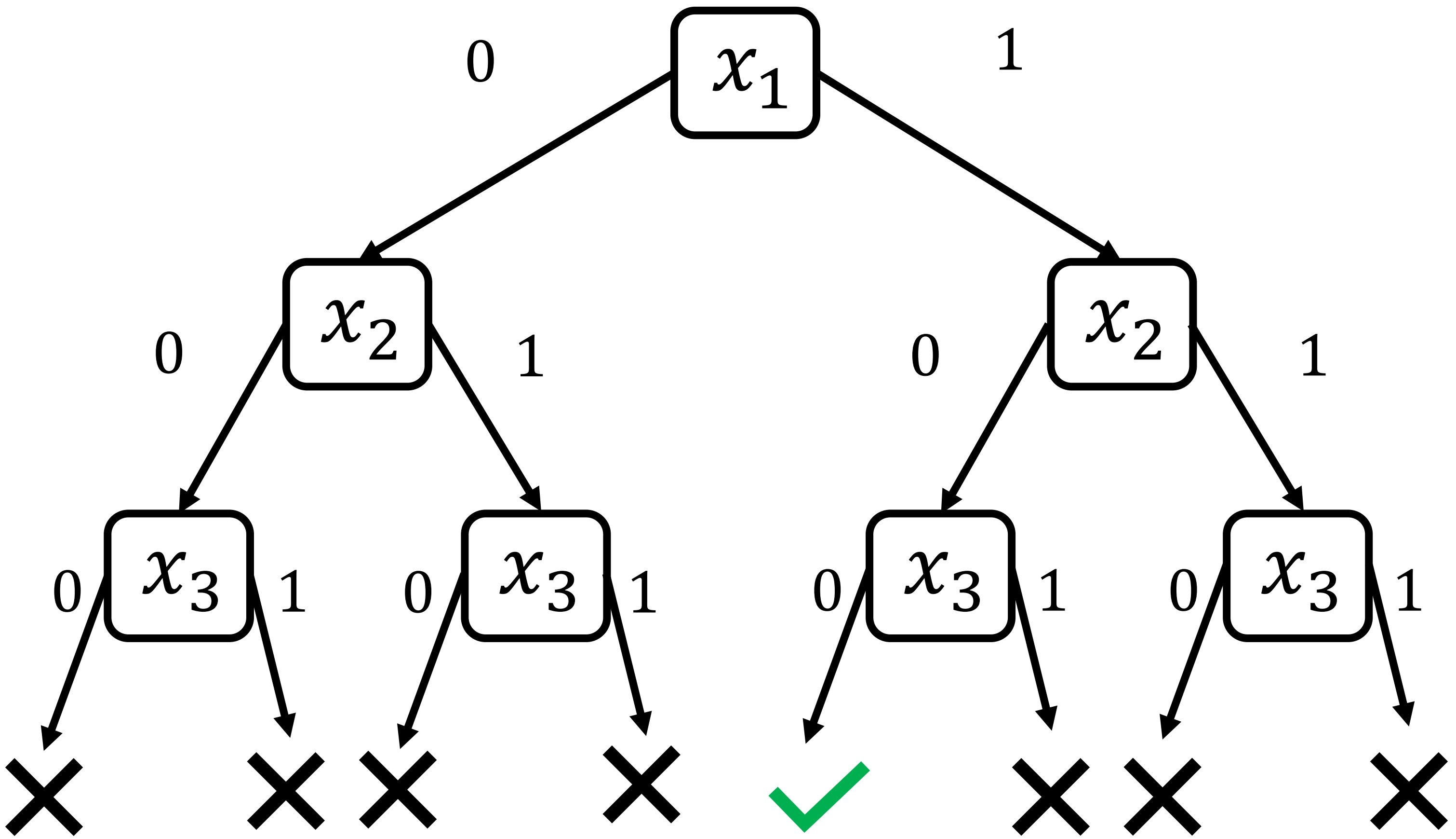


Search → depth-first search

in this course,
but other options possible

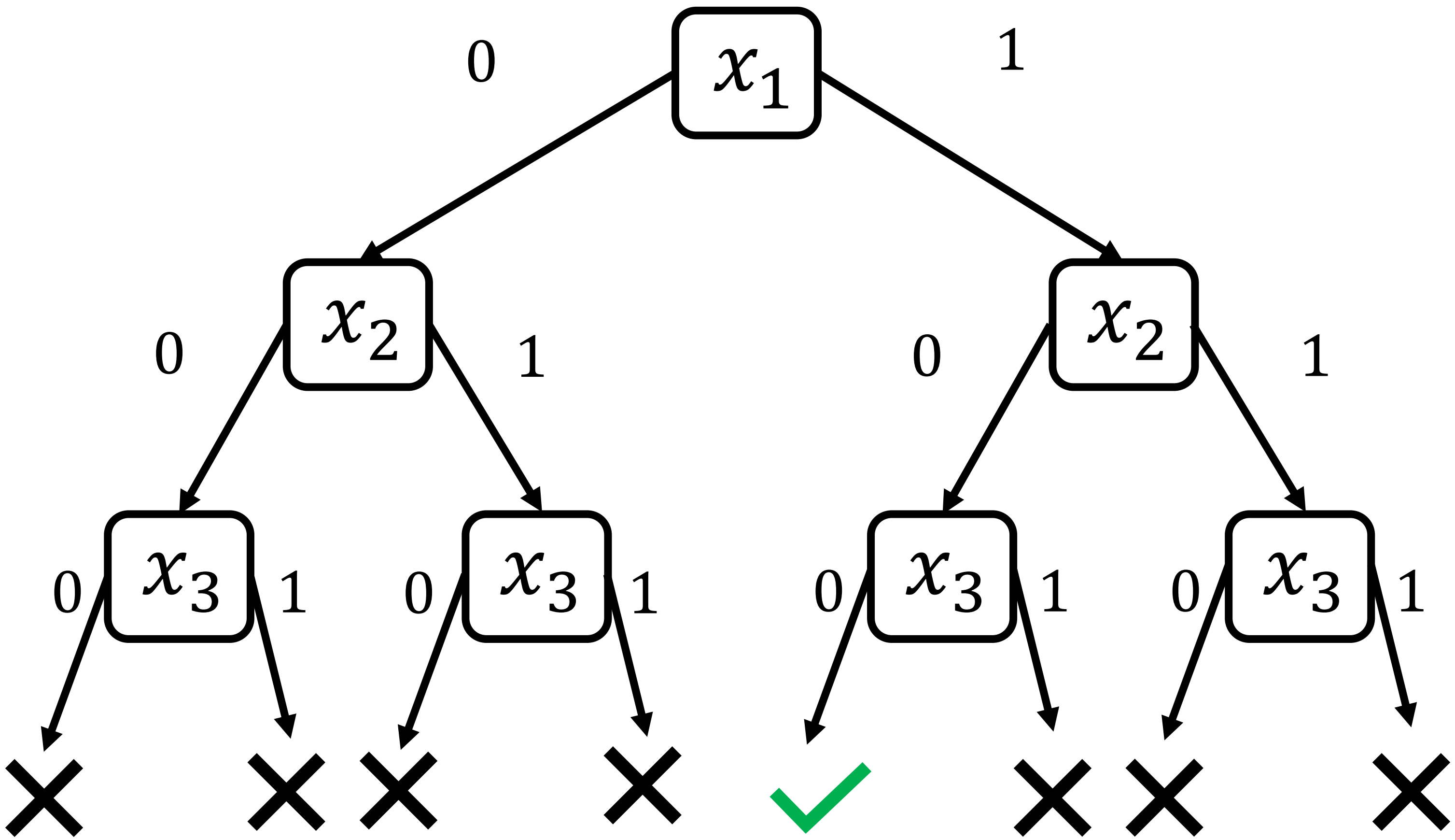


How large is the search tree?



How large is the search tree?

Exponential!



How many binary strings of length n?

How many binary strings of length n?

$\mathbf{2^n}$

number of assignments for n binary variables

Worst-case search tree size!

Exponential Growth

32-bit architecture

How many memory addresses?

Exponential Growth

32-bit architecture

How many memory addresses?

$$2^{32} \sim 4\ \text{GB}$$

Exponential Growth

32-bit architecture

How many memory addresses?

$$2^{32} \sim 4\ \mathbf{GB}$$

64-bit architecture

How many memory addresses?

Exponential Growth

32-bit architecture

How many memory addresses?

$$\mathbf{2^{32} \sim 4\ GB}$$

64-bit architecture

How many memory addresses?

$$\mathbf{2^{64} \sim 18,446,744,073\ GB}$$

Exponential Growth

32-bit architecture

How many memory addresses?

$$2^{32} \sim 4 \text{ GB}$$

64-bit architecture

How many memory addresses?

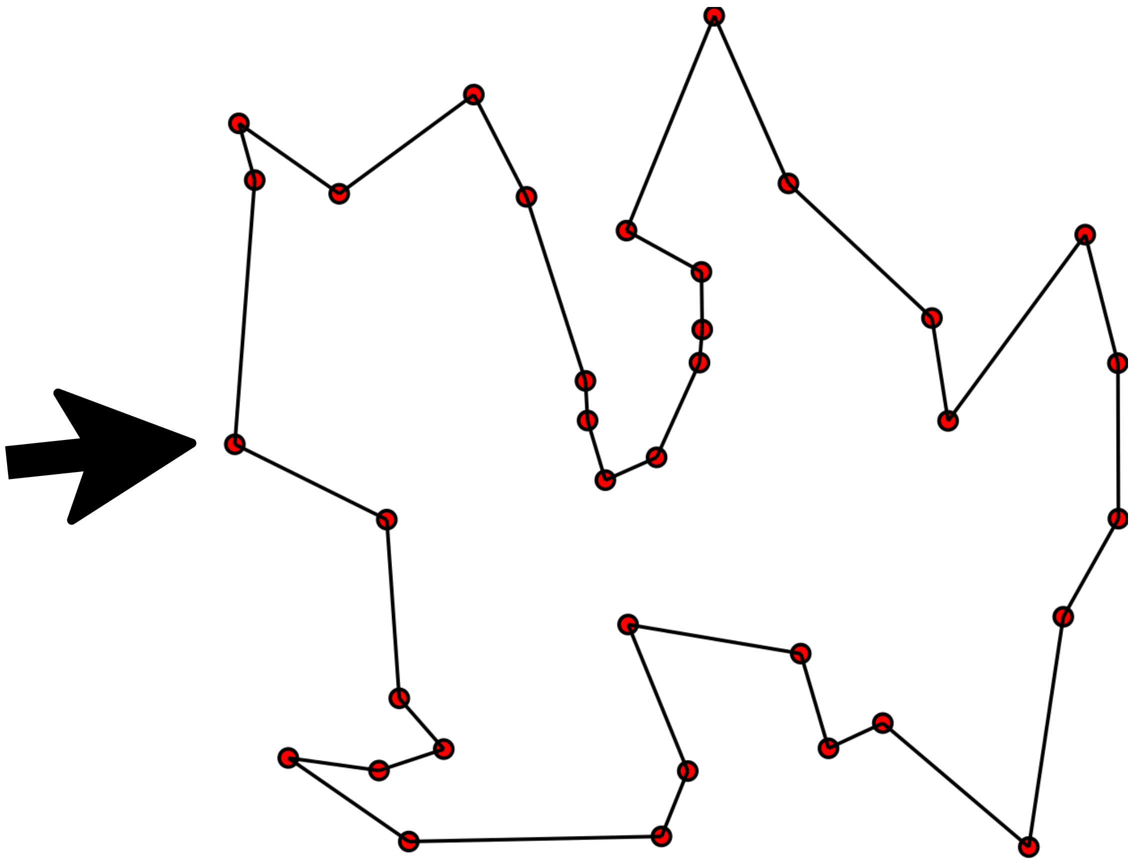
$$2^{64} \sim 18,446,744,073 \text{ GB}$$

Infinite for practical purposes!

Traveling Salesperson Problem

"Visit all nodes exactly once and return to the starting node"

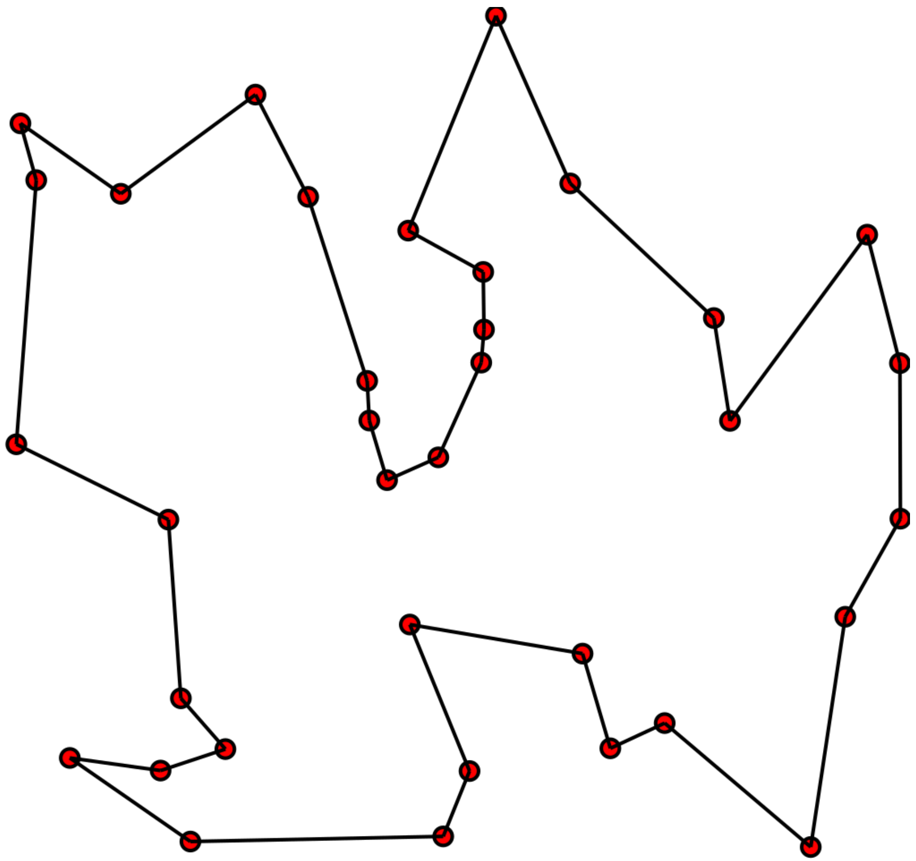
starting node



Traveling Salesperson Problem

"Visit all nodes exactly once and return to the starting node"

starting node

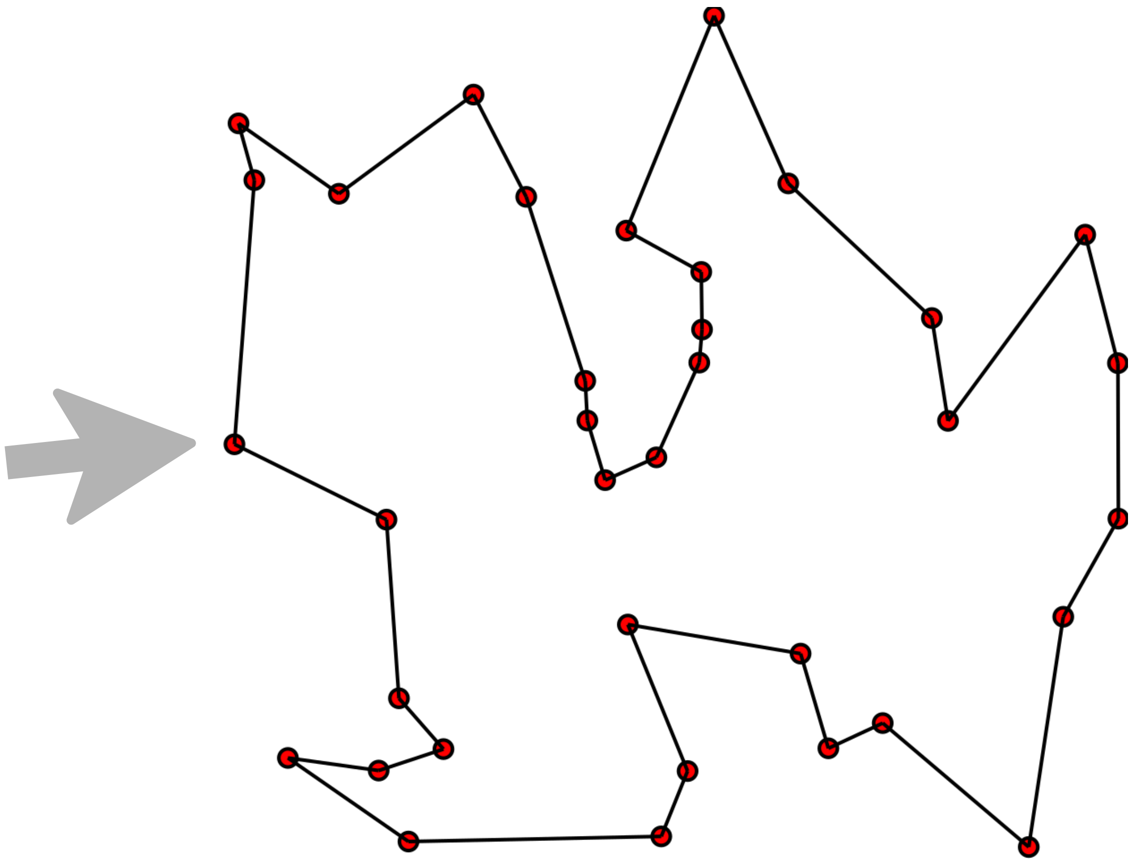


How many possible tours exist with 10 nodes?

Traveling Salesperson Problem

"Visit all nodes exactly once and return to the starting node"

starting node

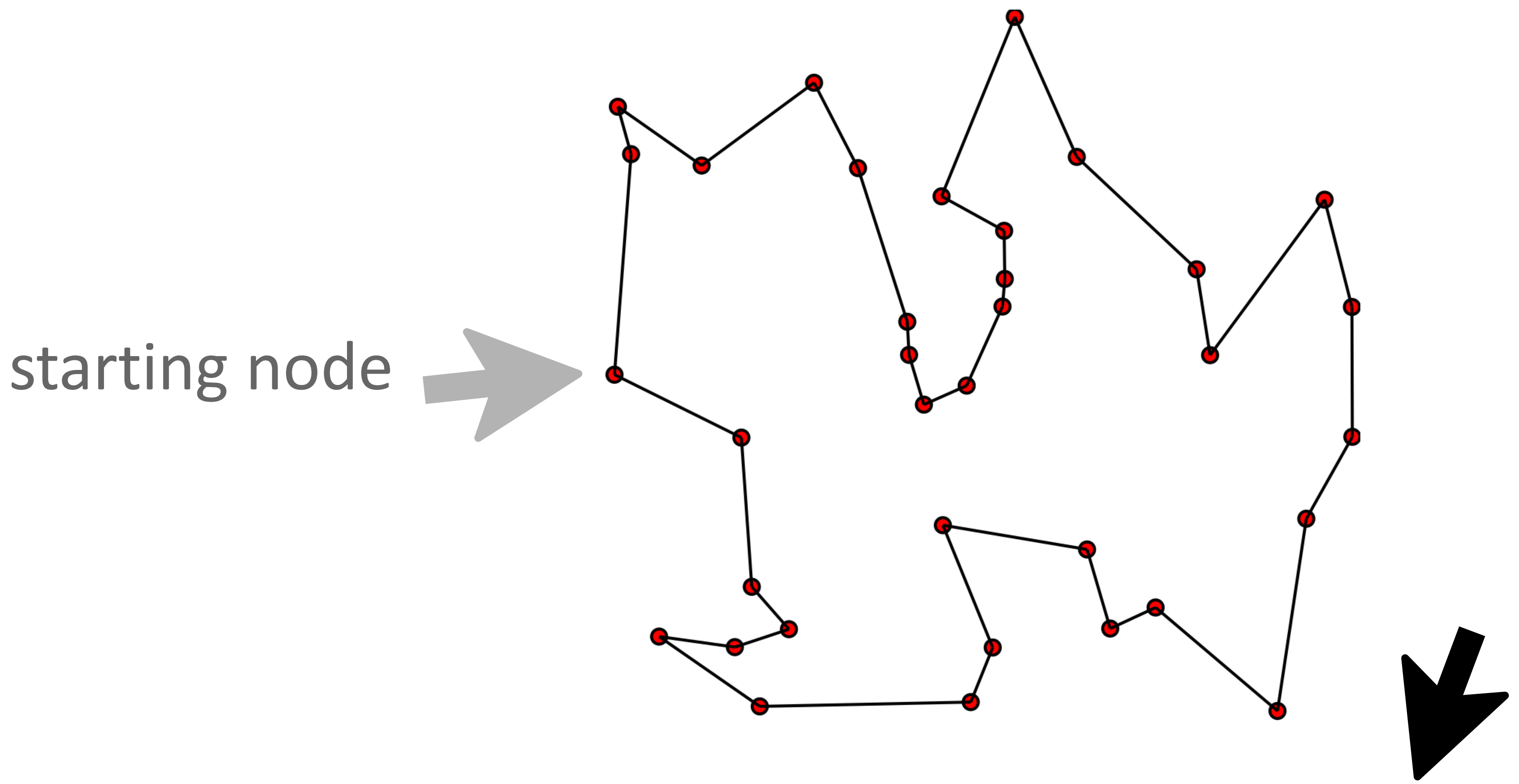


$\sim 10! = \sim 3{,}000{,}000$

How many possible tours exist with 10 nodes?

Traveling Salesperson Problem

"Visit all nodes exactly once and return to the starting node"

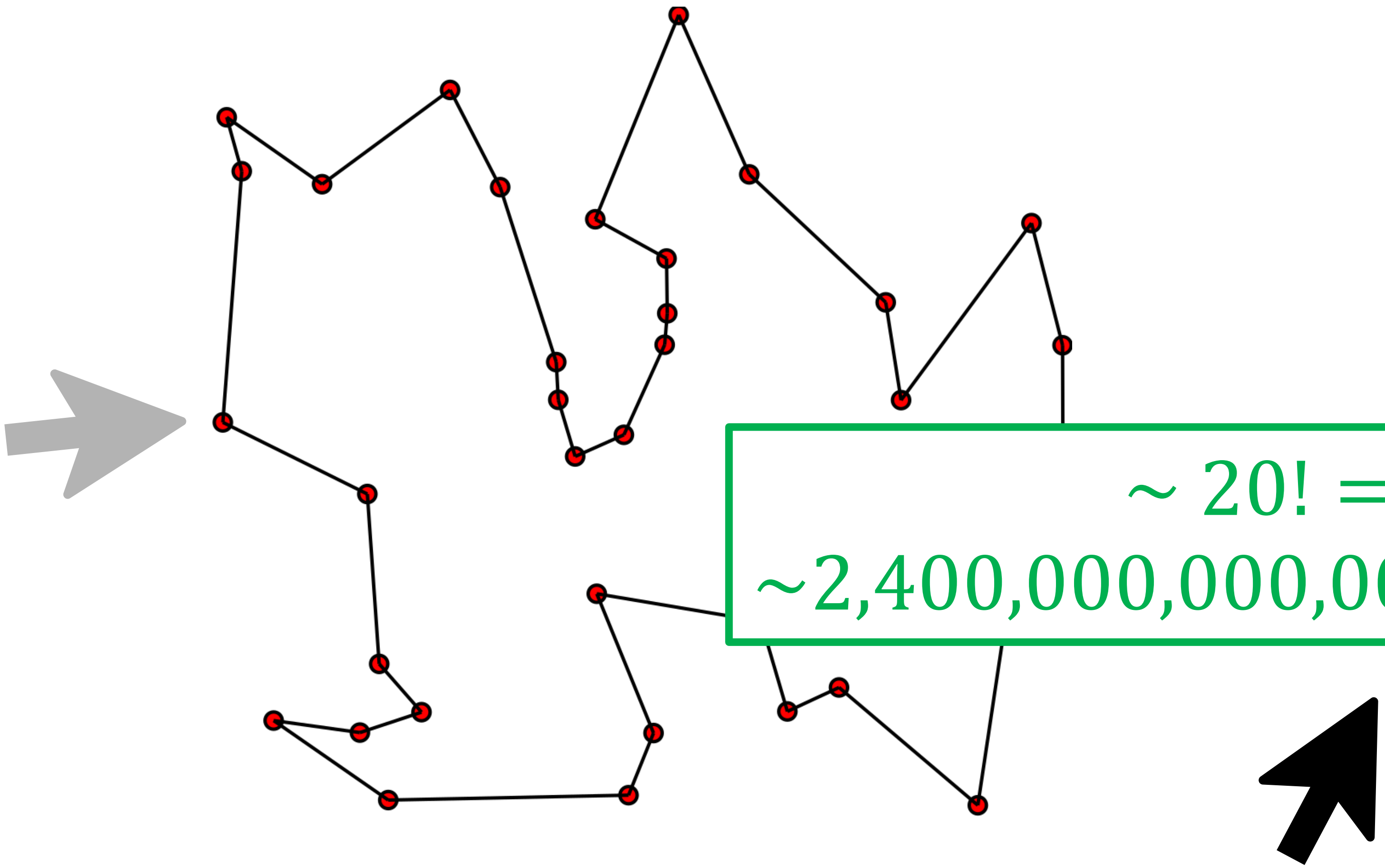


How many possible tours exist with 20 nodes?

Traveling Salesperson Problem

“Visit all nodes exactly once and return to the starting node”

starting node



$\sim 20! =$

$\sim 2{,}400{,}000{,}000{,}000{,}000{,}000$

How many possible tours exist with 20 nodes?

$$\sim 10! = \sim 3{,}000{,}000 = \sim 3 \cdot 10^6$$

$$\sim 20! = \sim 2{,}400{,}000{,}000{,}000{,}000{,}000 = \sim 2.4 \cdot 10^{18}$$

Given a magical unbreakable sheet of paper

How many times do you need to fold it to reach the moon?

Given a magical unbreakable sheet of paper

How many times do you need to fold it to reach the moon?

~ 45 times

Exponential growth is not intuitive!

Grows extremely quickly

…but the search tree size of combinatorial problems in practice easily exceeds 2^{1000}

Exponential growth is not intuitive!

Modelling and propagation are key!

Grows extremely quickly

...but the search tree size of combinatorial problems in practice easily exceeds 2^{1000}

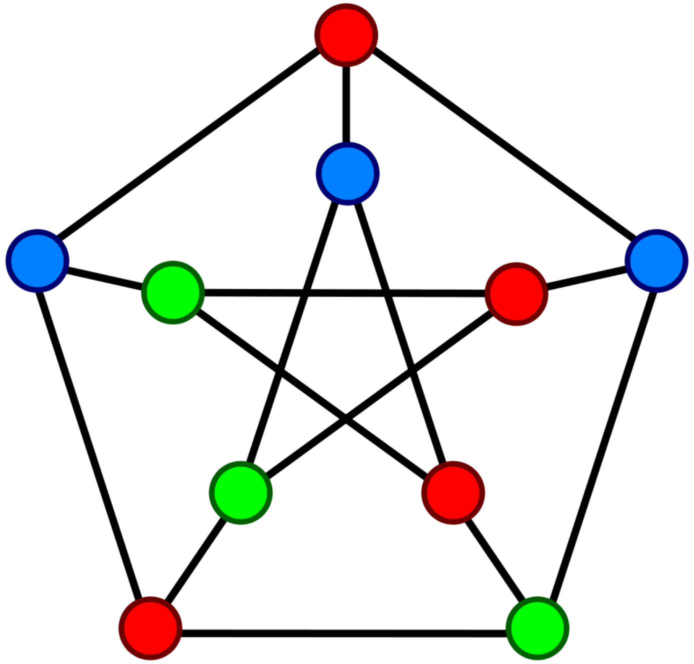
Many different problems share (similar) models!

Timetabling and graph colouring

Courses → nodes

Incompatibilities → edges

Time slots → colours



Modelling Patterns

Assignments

The i-*th* entity is assigned the j-*th* object

Modelling Patterns

Assignments

The i-th entity is assigned the j-th object

Node i assigned colour j

Task i assigned starting time j

Employee i assigned shift j

Modelling Patterns

Assignments

The *i-th* entity is assigned the *j-th* object

Option #1: Integer variable

$$Y_i \in \{1, 2, \ldots, n\}$$

$$Y_i = j$$

Node i assigned colour j

Task i assigned starting time j

Employee i assigned shift j

Modelling Patterns

Assignments

The i-th entity is assigned the j-th object

Option #2: Boolean variable

$$X_{i,j} \in \{0, 1\}$$

"is node i assigned colour j?"

Node i assigned colour j

Task i assigned starting time j

Employee i assigned shift j

Modelling Patterns

Assignments

The *i-th* entity is assigned the *j-th* object

Option #2: Boolean variable

$$X_{i,j} \in \{0, 1\}$$

Typically constraints needed to define the variables:

$$\forall i: \sum_j X_{i,j} = 1$$

"exactly one colour for each node"

$$X_{i,t,c} \in \{0, 1\}$$

"does lecturer i
at time t
teach course c?"

$$X_{i,t,c} \in \{0, 1\}$$

“does lecturer i
at time t
teach course c?”

How to model
“Lecturer can only teach on
at most two different days?”

$X_{i,t,c} \in \{0, 1\}$

"does lecturer i
at time t
teach course c?"

$M_{i,d} \in \{0, 1\}$

"does lecturer i teach on day d?"

$X_{i,t,c} \in \{0,1\}$

"does lecturer i
at time t
teach course c?"

$M_{i,d} \in \{0,1\}$

"does lecturer i teach on day d?"

Need constraints
to define variable!

$X_{i,t,c} \in \{0,1\}$

"does lecturer i
at time t
teach course c?"

$M_{i,d} \in \{0,1\}$

"does lecturer i teach on day d?"

$$\boxed{\begin{array}{c}\forall t \in times(d), \mathrm{c} \in courses(i): \\ X_{i,t,c} \rightarrow M_{i,d}\end{array}}$$

$X_{i,t,c} \in \{0,1\}$

"does lecturer i
at time t
teach course c?"

$M_{i,d} \in \{0,1\}$

"does lecturer i teach on day d?"

$\forall t \in times(d), \mathrm{c} \in courses(i)$:
$X_{i,t,c} \rightarrow M_{i,d}$

$$\boxed{M_{i,d} \rightarrow \bigvee_{t \in times(d), c \in courses(i)} X_{i,t,c}}$$

$X_{i,t,c} \in \{0,1\}$

"does lecturer i
at time t
teach course c?"

$M_{i,d} \in \{0,1\}$

"does lecturer i teach on day d?"

Constraints define the variable!

$$\forall t \in times(d), \text{c} \in courses(i): \quad X_{i,t,c} \rightarrow M_{i,d}$$

$$M_{i,d} \rightarrow \bigvee_{t \in times(d), c \in courses(i)} X_{i,t,c}$$

$X_{i,t,c} \in \{0,1\}$

"does lecturer i
at time t
teach course c?"

$M_{i,d} \in \{0,1\}$

"does lecturer i teach on day d?"

"Teacher can only teach on at most two different days"

$$\forall i: \sum_{d} M_{i,d} \leq 2$$

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

examples

"does lecturer i teach on day d?"

$$X_{i,t,c} \in \{0, 1\}$$

"does lecturer i
at time t
teach course c?"

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

$X_{i,t,c} \in \{0, 1\}$

"does lecturer i
at time t
teach course c?"

examples

"does lecturer i teach on day d?"

"does lecturer i have a schedule hole at time t?"

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

examples

$X_{i,t,c} \in \{0, 1\}$

“does lecturer i
at time t
teach course c?”

“does lecturer i teach on day d?”

“does lecturer i have a schedule hole at time t?”

“does lecturer i teach at time t?”

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

$$X_{i,t,c} \in \{0, 1\}$$

"does lecturer i
at time t
teach course c?"

examples

"does lecturer i teach on day d?"

"does lecturer i have a schedule hole at time t?"

"does lecturer i teach at time t?"

"is course assigned to lecturer i?"

"is course c taking place at time t?"

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

“Starting time for task i"

$$S_i \in \{0,1,\dots n\}$$

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

“Starting time for task i"

$$S_i \in \{0,1, \dots n\}$$

Auxiliary variable

“Is task i taking place at time t?”

$$X_{i,t} \in \{0,1\}$$

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

“Starting time for task i"

$S_i \in \{0,1,\dots n\}$

Auxiliary variable

“Is task i taking place at time t?”

$X_{i,t} \in \{0,1\}$

“cannot exceed resource capacity”

$$\forall t: \sum_i r_i \cdot X_{i,t} \leq R_{max}$$

Modelling Patterns

Auxiliary Variables

Define new variables to help model constraints

“Starting time for task i"

$S_i \in \{0,1, \dots n\}$

Auxiliary variable

“Is task i taking place at time t?”

$X_{i,t} \in \{0,1\}$

“cannot exceed resource capacity”

$$\forall t\colon \sum_i r_i \cdot X_{i,t} \leq R_{max}$$

$$\boxed{\text{Auxiliary variables are useful in modelling!}}$$

Modelling Patterns

Step Functions

$$y = f(x) = \max\{x - k, 0\}$$

Modelling Patterns

Step Functions

$$y = f(x) = \max\{x - k, 0\}$$

“all duties preferably should fit within two days”

Modelling Patterns

Step Functions

$$y = f(x) = \max\{x - k, 0\}$$

“all duties preferably should fit within two days”

“resource consumption preferably should not exceed 20 units”

Modelling Patterns
Step Functions

$$y = f(x) = \max\{x - k, 0\}$$

“all duties preferably should fit within two days”

“resource consumption preferably should not exceed 20 units”

“lecturers preferably should not have more than two holes in their schedule”

Modelling Patterns

Step Functions

$$y = f(x) = \max\{x - k, 0\}$$

Typically used in the objective function!

Modelling Patterns

Assignments

Auxiliary Variables

Step Functions

Modelling Patterns

Assignments

Auxiliary Variables

Step Functions

High-level constraints

Global Constraints

$$x_1, x_2, x_3, x_4 \in \{1, 2, 3, 4\}$$

"all variables must take distinct values"

How to model?

$$x_1, x_2, x_3, x_4 \in \{1, 2, 3, 4\}$$

"all variables must take distinct values"

$$x_i \neq x_j$$

$$\boxed{\begin{array}{lll} x_1 \neq x_2 & & \\ x_1 \neq x_3 & x_2 \neq x_3 & \\ x_1 \neq x_4 & x_2 \neq x_4 & x_3 \neq x_4 \end{array}}$$

$$x_1 \in \{1, 3\}$$
$$x_2 \in \{1, 3\}$$
$$x_3 \in \{1, 2, 4\}$$
$$x_4 \in \{1, 2\}$$

$$x_1 \neq x_2$$

$$x_1 \neq x_3 \qquad x_2 \neq x_3$$

$$x_1 \neq x_4 \qquad x_2 \neq x_4 \qquad x_3 \neq x_4$$

$$x_1 \in \{1,3\}$$
$$x_2 \in \{1,3\}$$
$$x_3 \in \{1,2,4\}$$
$$x_4 \in \{1,2\}$$

no propagation!

$x_1 \neq x_2$

$x_1 \neq x_3$ $\quad$ $x_2 \neq x_3$

$x_1 \neq x_4$ $\quad$ $x_2 \neq x_4$ $\quad$ $x_3 \neq x_4$

$$x_1 \in \{1, 3\}$$
$$x_2 \in \{1, 3\}$$
$$x_3 \in \{1, 2, 4\}$$
$$x_4 \in \{1, 2\}$$

'4' must be assigned here

$$x_1 \neq x_2$$

$$x_1 \neq x_3 \qquad x_2 \neq x_3$$

$$x_1 \neq x_4 \qquad x_2 \neq x_4 \qquad x_3 \neq x_4$$

$x_1 \in \{1, 3\}$

$x_2 \in \{1, 3\}$

$x_3 \in \{4\}$

$x_4 \in \{1, 2\}$ ← **'2' must be assigned here**

$$\boxed{\begin{array}{lll} x_1 \neq x_2 & & \\ x_1 \neq x_3 & x_2 \neq x_3 & \\ x_1 \neq x_4 & x_2 \neq x_4 & x_3 \neq x_4 \end{array}}$$

$$x_1 \in \{1, 3\}$$
$$x_2 \in \{1, 3\}$$
$$x_3 \in \{4\}$$
$$x_4 \in \{2\}$$

$$x_1 \neq x_2$$

$$x_1 \neq x_3 \qquad x_2 \neq x_3$$

$$x_1 \neq x_4 \qquad x_2 \neq x_4 \qquad x_3 \neq x_4$$

$x_1 \in \{1, 3\}$
$x_2 \in \{1, 3\}$
$x_3 \in \{4\}$
$x_4 \in \{2\}$

Stronger reasoning by combining constraints!

$$\boxed{\begin{array}{lll} x_1 \neq x_2 & & \\ x_1 \neq x_3 & x_2 \neq x_3 & \\ x_1 \neq x_4 & x_2 \neq x_4 & x_3 \neq x_4 \end{array}}$$

$$x_1 \in \{1, 2, 3\}$$
$$x_2 \in \{1, 2, 3\}$$
$$x_3 \in \{1, 2, 3\}$$
$$x_4 \in \{1, 2\}$$

$$x_1 \neq x_2$$

$$x_1 \neq x_3 \qquad x_2 \neq x_3$$

$$x_1 \neq x_4 \qquad x_2 \neq x_4 \qquad x_3 \neq x_4$$

$$x_1 \in \{1, 2, 3\}$$
$$x_2 \in \{1, 2, 3\}$$
$$x_3 \in \{1, 2, 3\}$$
$$x_4 \in \{1, 2\}$$

no propagation even though clearly infeasible!

$$\boxed{\begin{array}{lll} x_1 \neq x_2 & & \\ x_1 \neq x_3 & x_2 \neq x_3 & \\ x_1 \neq x_4 & x_2 \neq x_4 & x_3 \neq x_4 \end{array}}$$

$$x_1 \in \{1, 2, 3\}$$
$$x_2 \in \{1, 2, 3\}$$
$$x_3 \in \{1, 2, 3\}$$
$$x_4 \in \{1, 2\}$$

no propagation
even though
clearly infeasible!

$$x_1 \neq x_2$$

$$x_1 \neq x_3 \qquad x_2 \neq x_3$$

$$x_1 \neq x_4 \qquad x_2 \neq x_4 \qquad x_3 \neq x_4$$

Global Constraints

All-Different

"all variables must take distinct values"

All-Different(x_1, x_2, x_3, x_4)

Global Constraints

All-Different

"all variables must take distinct values"

All-Different(x_1, x_2, x_3, x_4)

Common in modelling assignments!

Global Constraints

All-Different

"all variables must take distinct values"

All-Different(x_1, x_2, x_3, x_4)

$$\boxed{\begin{aligned} x_1 &= 1 \\ x_2 &= 2 \\ x_3 &= 5 \\ x_4 &= 3 \end{aligned}}$$

Global Constraints

All-Different

"all variables must take distinct values"

All-Different(x_1, x_2, x_3, x_4)

$$x_1 = 1 \quad x_2 = 2 \quad x_3 = 5 \quad x_4 = 3$$

$$x_1 = 1 \quad x_2 = 1 \quad x_3 = 5 \quad x_4 = 3$$

Global Constraints

All-Different

$$x_i \in D_i \subset \mathbb{N}$$

$$\text{All-Different}(x_1, x_2, \ldots, x_n)$$

How to determine feasibility?

Global Constraints

All-Different

variables

x_1 1

x_2 2

x_3 3

x_4 4

Global Constraints

All-Different

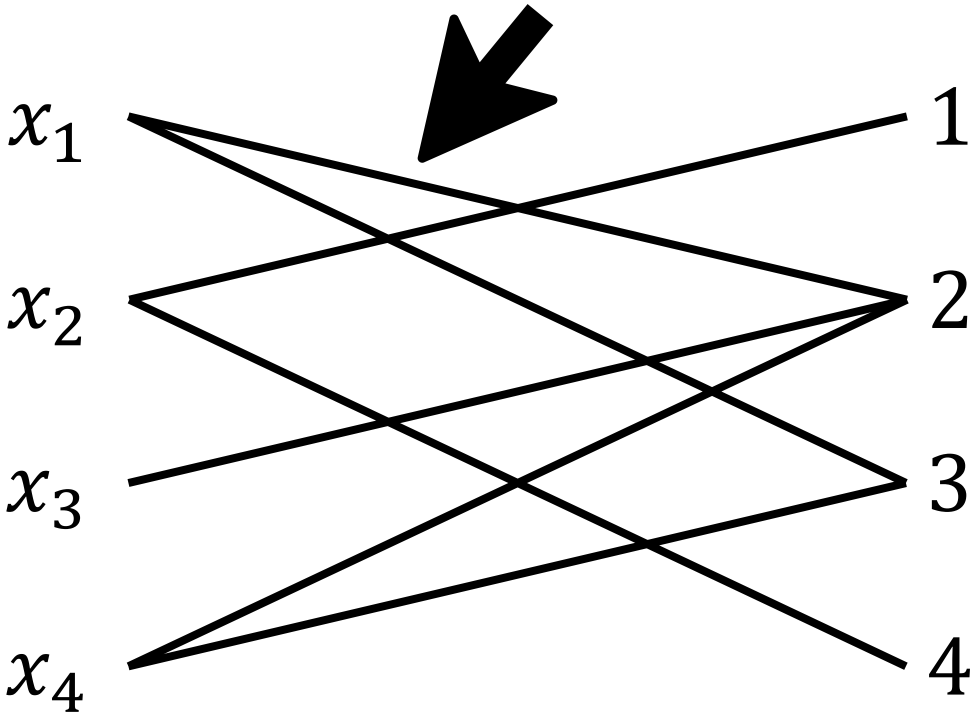
Domain values

x_1	1
x_2	2
x_3	3
x_4	4

Global Constraints

All-Different

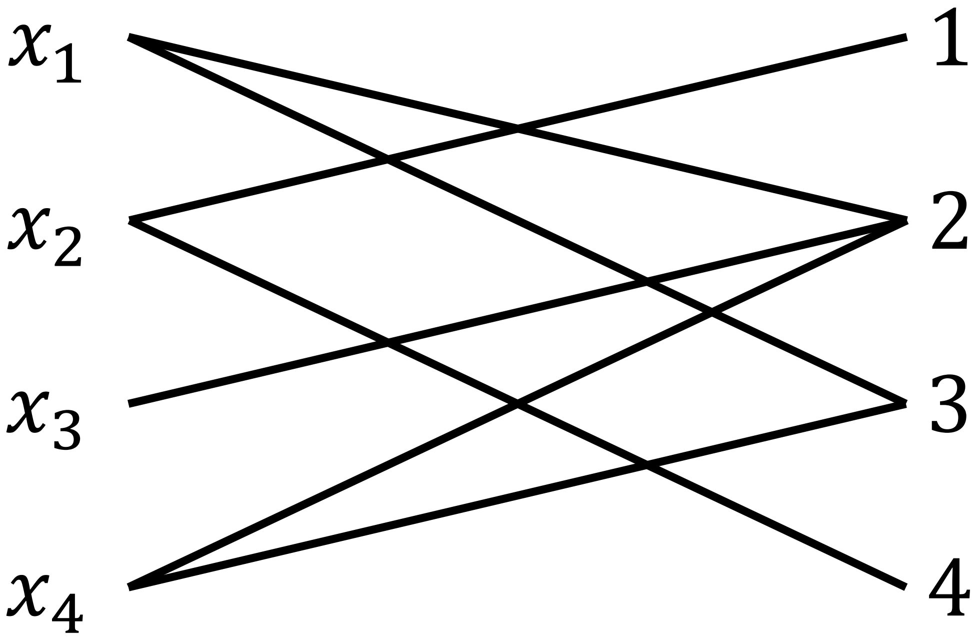
edge connect variable to domain value



Global Constraints

All-Different

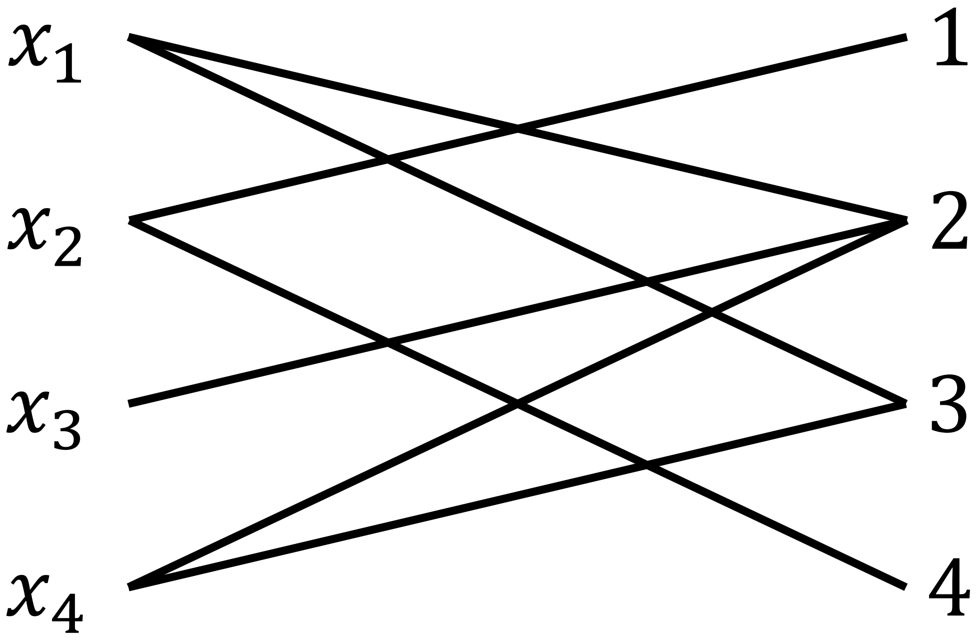
Bi-partite graph



Global Constraints

All-Different

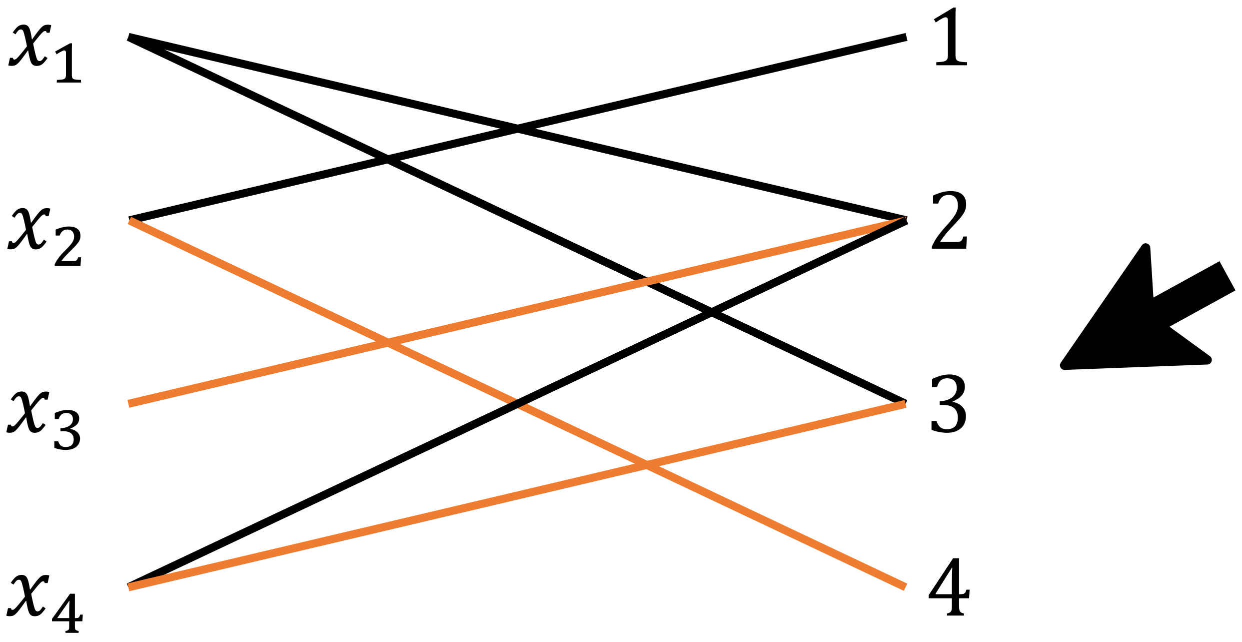
“all-different is feasible if there exists a matching of size four”



Global Constraints

All-Different

Maximum matching is three, infeasible!



The global constraint All-Different useful for both…

Modelling

Propagation

Many different global constraints

Disjunctive$(s_1, s_2, \dots, s_n, L_1, L_2, \dots, L_n)$

"ensure tasks do not overlap"

Many different global constraints

$\text{Disjunctive}(s_1, s_2, \ldots, s_n, L_1, L_2, \ldots, L_n)$

s_i - integer variable, starting time for task i

Many different global constraints

Disjunctive$(s_1, s_2, \ldots, s_n, L_1, L_2, \ldots, L_n)$

L_i - constant, duration of task i

Many different global constraints

Disjunctive$(s_1, s_2, \ldots, s_n, L_1, L_2, \ldots, L_n)$

"ensure tasks do not overlap"

Many different global constraints

Disjunctive$(s_1, s_2, \ldots, s_n, L_1, L_2, \ldots, L_n)$

"ensure tasks do not overlap"

Cumulative$(s_1, s_2, \ldots, s_n, L_1, L_2, \ldots, L_n, r_1, r_2, \ldots, r_n, R_{max})$

r_i - constant, resource requirements for task i

Many different global constraints

Disjunctive$(s_1, s_2, \dots, s_n, L_1, L_2, \dots, L_n)$

"ensure tasks do not overlap"

Cumulative$(s_1, s_2, \dots, s_n, L_1, L_2, \dots, L_n, r_1, r_2, \dots, r_n, R_{max})$

Num resources available

Many different global constraints

Disjunctive$(s_1, s_2, \ldots, s_n, L_1, L_2, \ldots, L_n)$

"ensure tasks do not overlap"

Cumulative$(s_1, s_2, \ldots, s_n, L_1, L_2, \ldots, L_n, r_1, r_2, \ldots, r_n, R_{max})$

"ensure tasks do not exceed maximum capacity"

Many different global constraints

Linear inequality
Disjunctive
Cumulative
All-Different
Circuit
Element
Table
Difference Logic
Regular
…

https://www.minizinc.org/doc-2.5.5/en/lib-globals.html
http://sofdem.github.io/gccat/

→ over 300 global constraints listed!

Many different global constraints

Stronger reasoning by exploiting the structure!

Linear inequality
Disjunctive
Cumulative
All-Different
Circuit
Element
Table
Difference Logic
Regular
...

https://www.minizinc.org/doc-2.5.5/en/lib-globals.html
http://sofdem.github.io/gccat/

over 300 global constraints listed!

Modelling Patterns

Assignments

Auxiliary Variables

Step Functions

Global Constraints

$$X \subseteq \{0, 1, 2, 3, 4, 5, 6, 7, 8,9\}$$

$$|X| = 3$$

How to model a set of fixed cardinality?

$$X \subseteq \{0, 1, 2, 3, 4, 5, 6, 7, 8,9\}$$

$$|X| = 3$$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\text{All-Different}(x_1, x_2, x_3)$$

Model set with integer variables!

$$X \subseteq \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$|X| = 3$$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\text{All-Different}(x_1, x_2, x_3)$$

$$\{1, 4, 6\}$$

$$x_1 = 1$$
$$x_2 = 4$$
$$x_3 = 6$$

$$X \subseteq \{0, 1, 2, 3, 4, 5, 6, 7, 8,9\}$$

$$|X| = 3$$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

All-Different(x_1, x_2, x_3)

$$\{5, 0, 9\}$$

$$\begin{array}{lll} x_1 = 1 & \qquad x_1 = 5 \\ x_2 = 4 & \qquad x_2 = 0 \\ x_3 = 6 & \qquad x_3 = 9 \end{array}$$

$$X \subseteq \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$|X| = 3$$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

All-Different(x_1, x_2, x_3)

$\{1, 6, 4\}$

$x_1 = 1$	$x_1 = 5$	$x_1 = 1$
$x_2 = 4$	$x_2 = 0$	$x_2 = 6$
$x_3 = 6$	$x_3 = 9$	$x_3 = 4$

$$X \subseteq \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$|X| = 3$$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

All-Different(x_1, x_2, x_3)

$\{1, 4, 6\}$

$\{1, 6, 4\}$

$x_1 = 1$	$x_1 = 5$	$x_1 = 1$
$x_2 = 4$	$x_2 = 0$	$x_2 = 6$
$x_3 = 6$	$x_3 = 9$	$x_3 = 4$

Symmetry

Different variable assignments represent the same subset!

$$\{1,4,6\} \quad \begin{array}{l} x_1 = 1 \\ x_2 = 4 \\ x_3 = 6 \end{array} \qquad \begin{array}{l} x_1 = 5 \\ x_2 = 0 \\ x_3 = 9 \end{array} \qquad \begin{array}{l} x_1 = 1 \\ x_2 = 6 \\ x_3 = 4 \end{array} \quad \{1,6,4\}$$

Symmetry

Different variable assignments represent the same subset!

Unnecessarily increases the search space!

$\{1, 4, 6\}$ $x_1 = 1$, $x_2 = 4$, $x_3 = 6$

$x_1 = 5$, $x_2 = 0$, $x_3 = 9$

$x_1 = 1$, $x_2 = 6$, $x_3 = 4$ $\{1, 6, 4\}$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\text{All-Different}(x_1, x_2, x_3)$$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\text{All-Different}(x_1, x_2, x_3)$$

Symmetry breaking constraints →

$$x_1 < x_2$$
$$x_2 < x_3$$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\text{All-Different}(x_1, x_2, x_3)$$

$$x_1 < x_2$$
$$x_2 < x_3$$

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 4 \\ x_3 &= 6 \end{aligned}$$

$$\cancel{\begin{aligned} x_1 &= 5 \\ x_2 &= 0 \\ x_3 &= 9 \end{aligned}}$$

$$\cancel{\begin{aligned} x_1 &= 1 \\ x_2 &= 6 \\ x_3 &= 4 \end{aligned}}$$

$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

All-Different(x_1, x_2, x_3)

Each subset now has a unique representation!

$$x_1 < x_2$$
$$x_2 < x_3$$

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 4 \\ x_3 &= 6 \end{aligned} \qquad \cancel{\begin{aligned} x_1 &= 5 \\ x_2 &= 0 \\ x_3 &= 9 \end{aligned}} \qquad \cancel{\begin{aligned} x_1 &= 1 \\ x_2 &= 6 \\ x_3 &= 4 \end{aligned}}$$

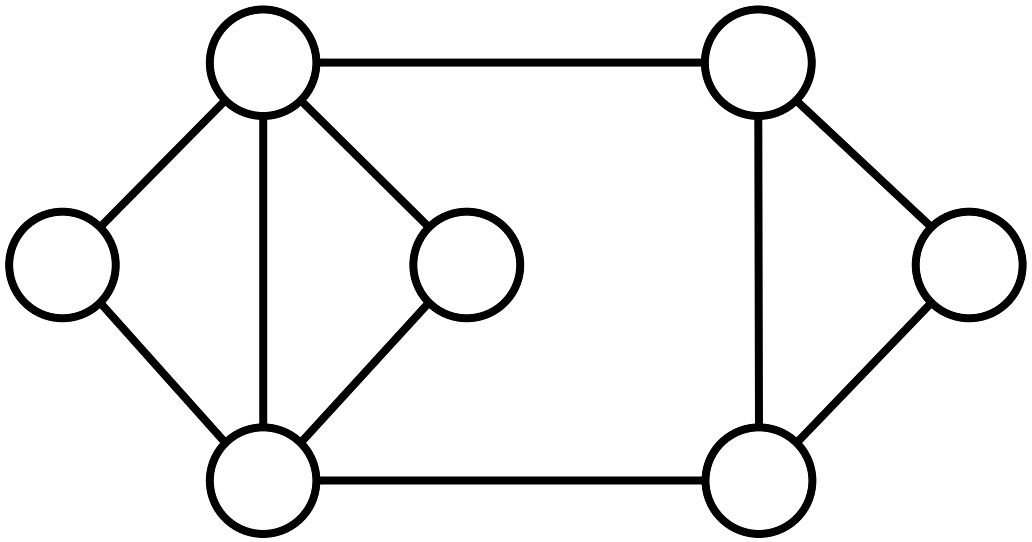
$$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

All-Different(x_1, x_2, x_3)

$$x_1 < x_2$$
$$x_2 < x_3$$

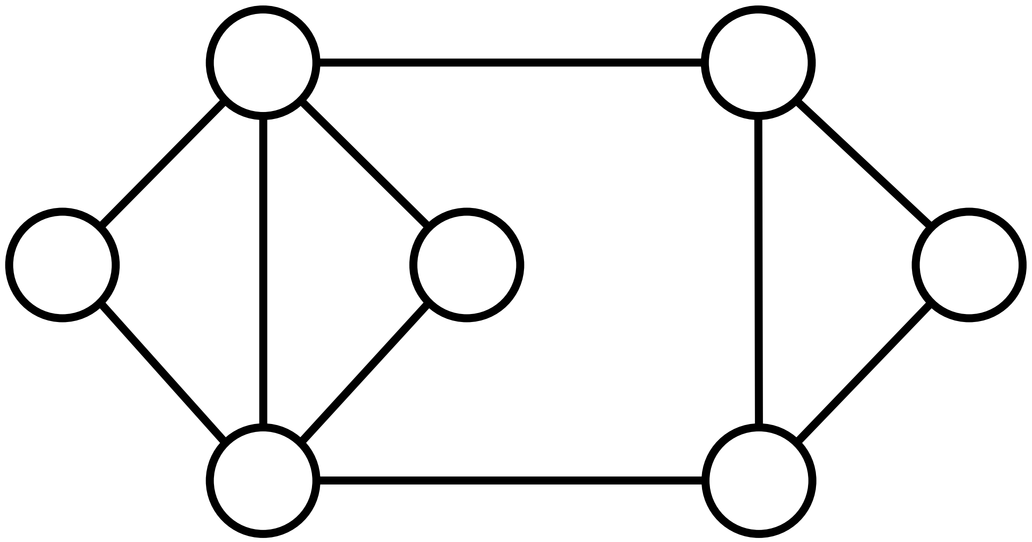
Each subset now has a unique representation!

$$\begin{array}{lll} x_1 = 1 & \cancel{x_1 = 5} & \cancel{x_1 = 1} \\ x_2 = 4 & \cancel{x_2 = 0} & \cancel{x_2 = 6} \\ x_3 = 6 & \cancel{x_3 = 9} & \cancel{x_3 = 4} \end{array}$$



Colour nodes

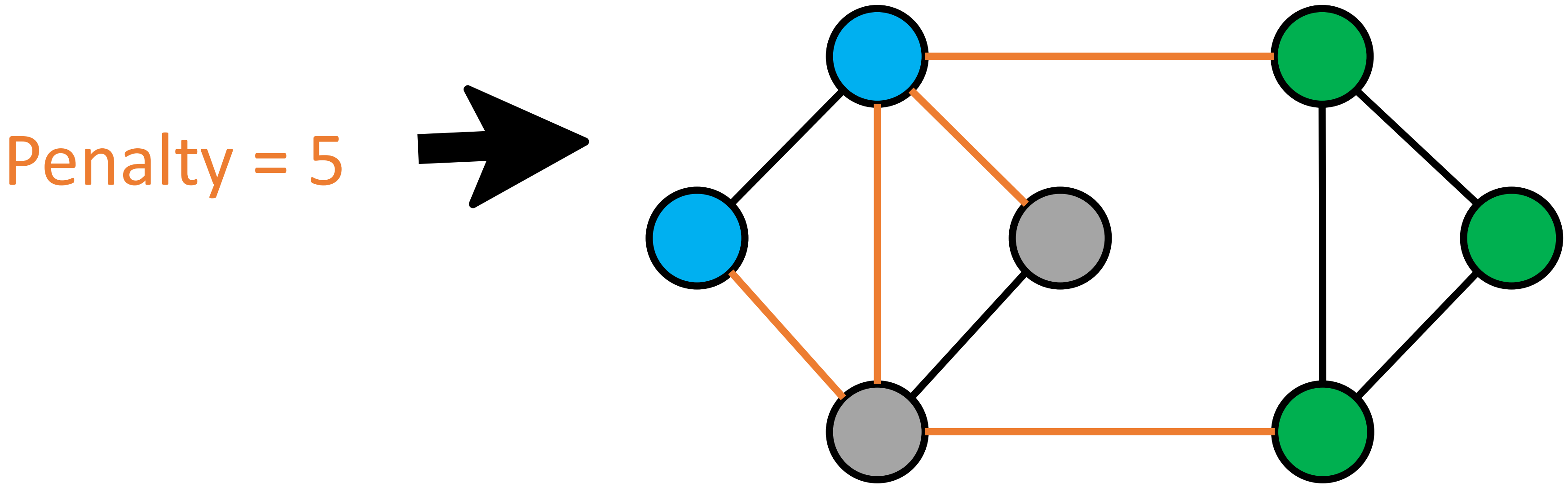
Connected nodes preferably in same colour



Colour nodes

Connected nodes preferably in same colour

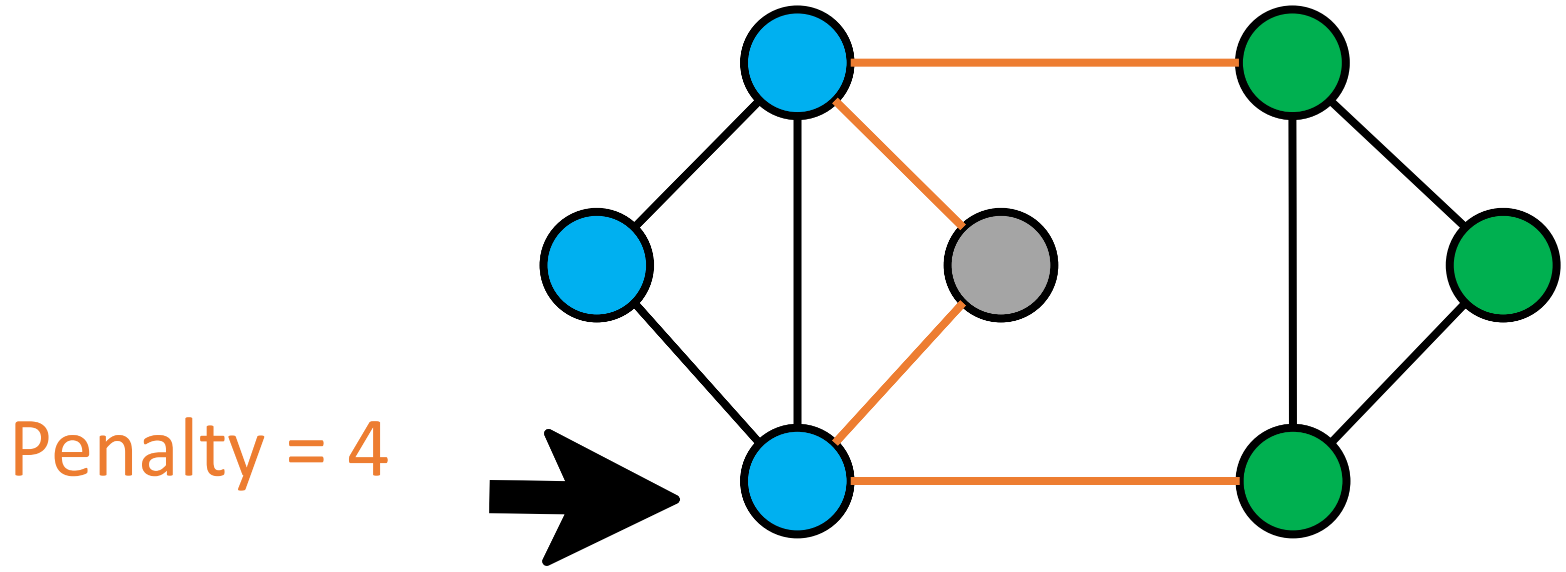
Disconnected nodes preferably in different colours



Colour nodes

Connected nodes preferably in same colour

Disconnected nodes preferably in different colours



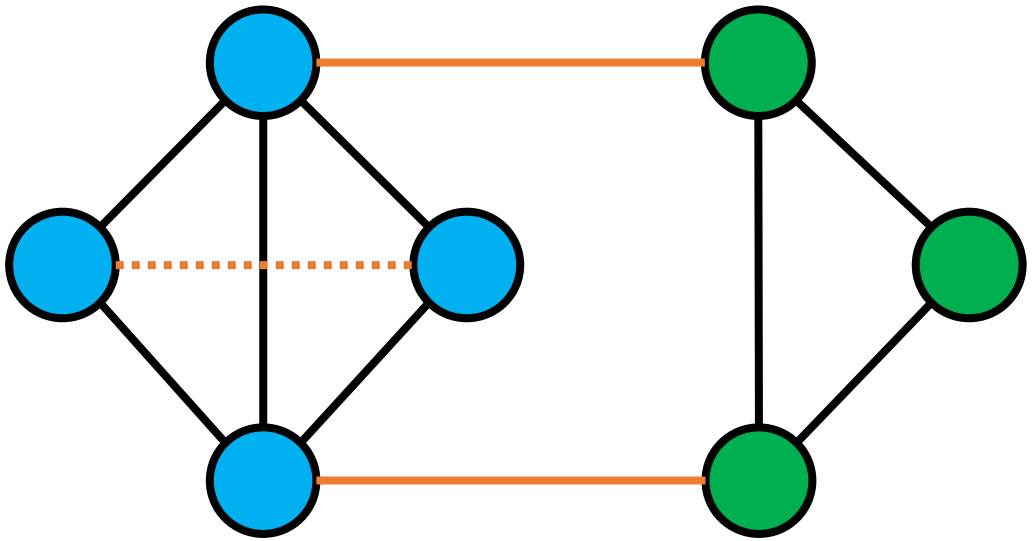
Colour nodes

Connected nodes preferably in same colour

Disconnected nodes preferably in different colours

How to model?

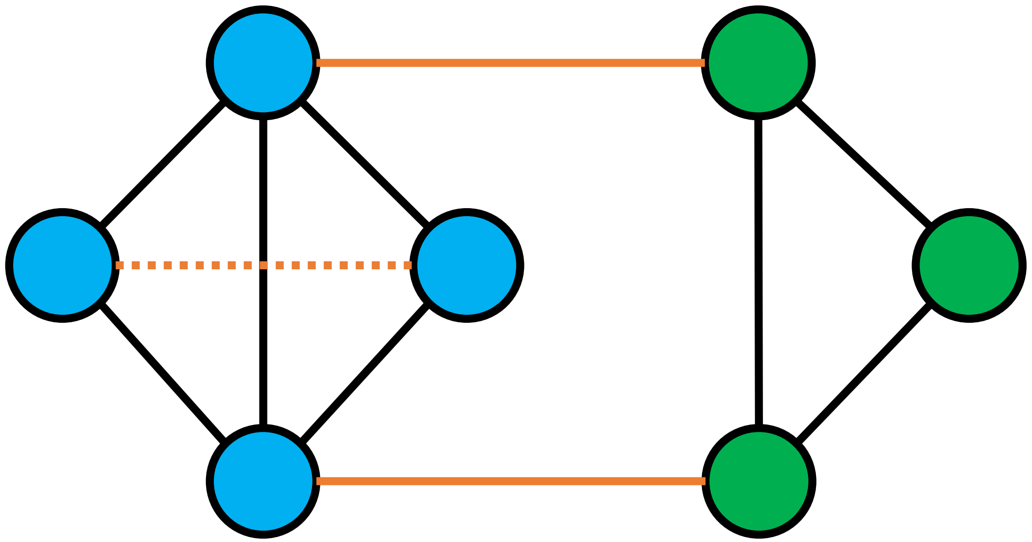
Penalty = 3



Colour nodes

Connected nodes preferably in same colour

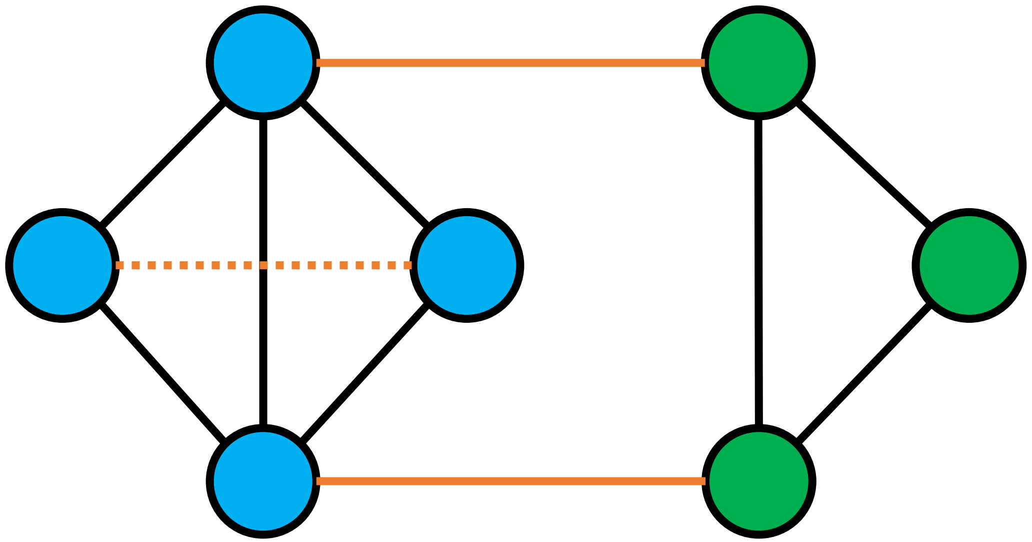
Disconnected nodes preferably in different colours



Option #1

$$x_i = \{1, 2, 3, \ldots, n\}$$

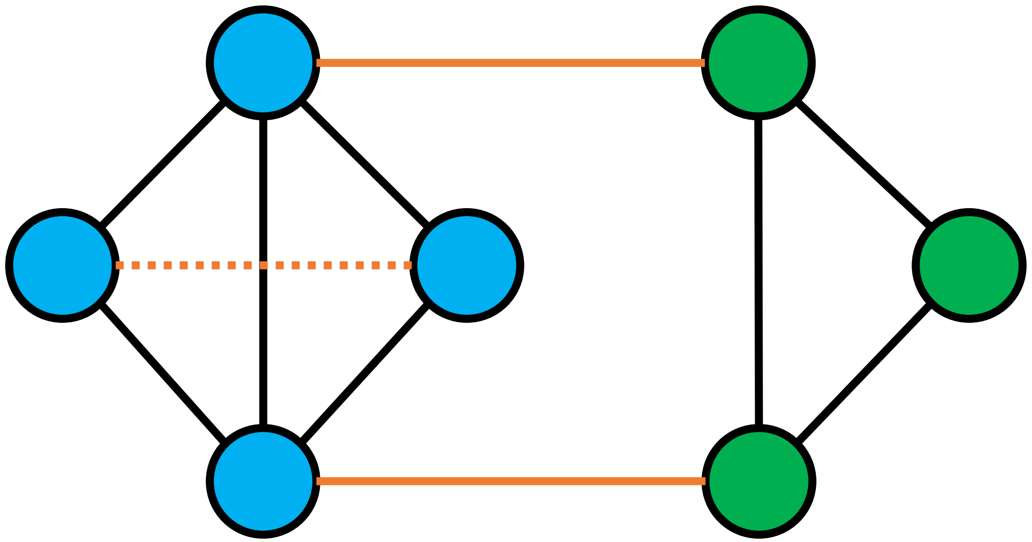
"x_i is the colour of node i"



Option #1

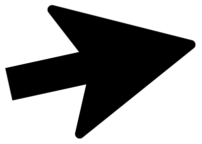
$$x_i = \{1, 2, 3, \ldots, n\}$$

Where is the symmetry?

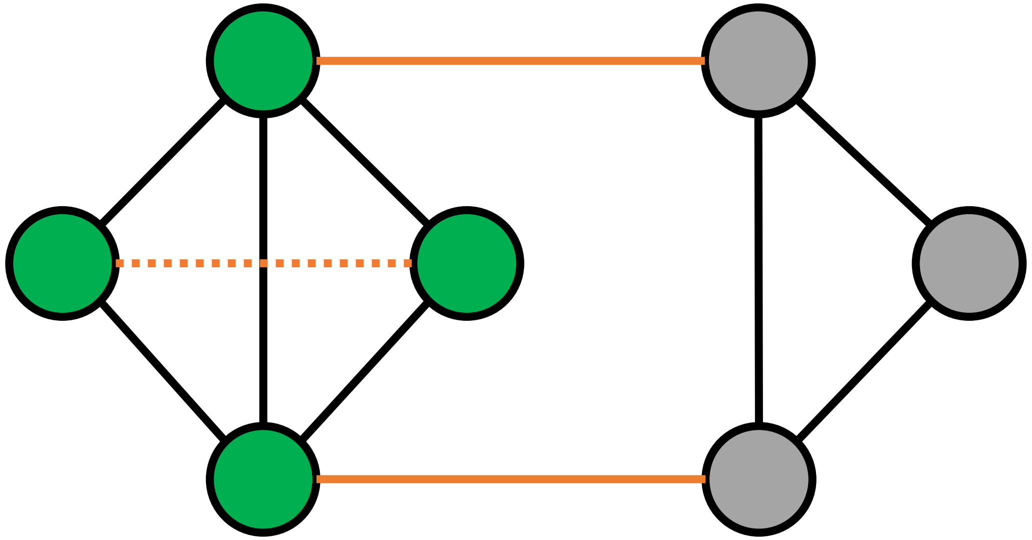


Option #1

$$x_i = \{1, 2, 3, \ldots, n\}$$

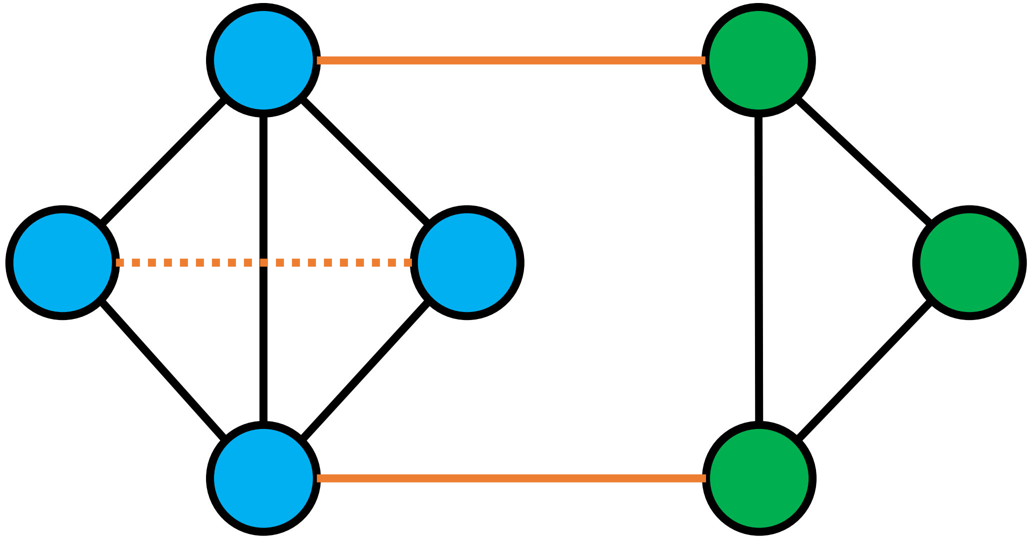


Colours can be renamed!

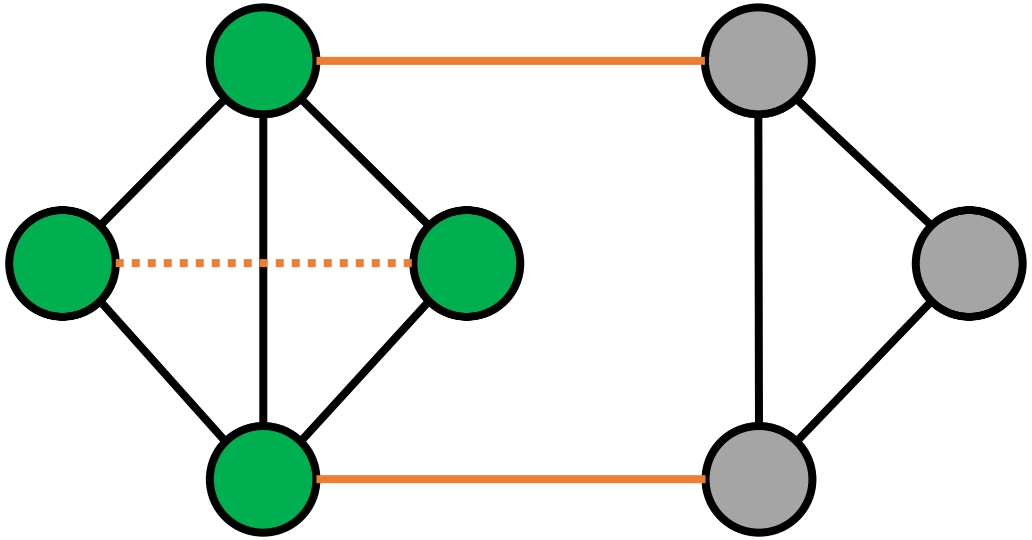
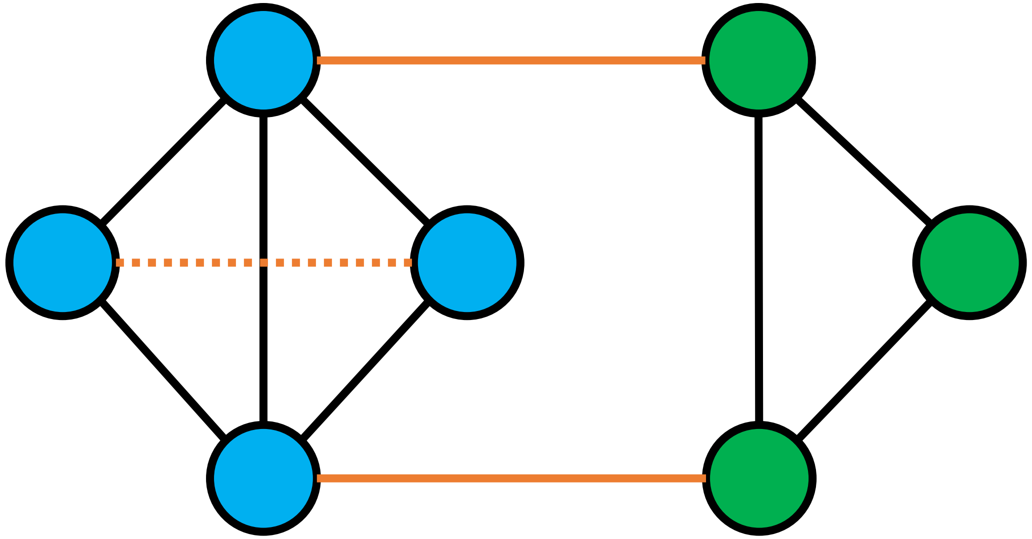


Option #1

$$x_i = \{1, 2, 3, \dots, n\}$$



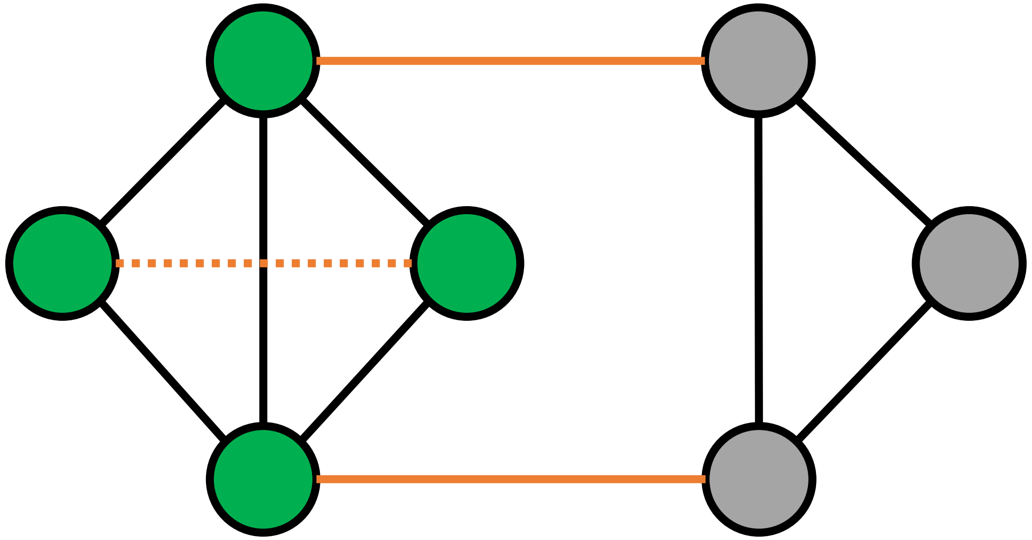
Two symmetric solutions!

Option #1

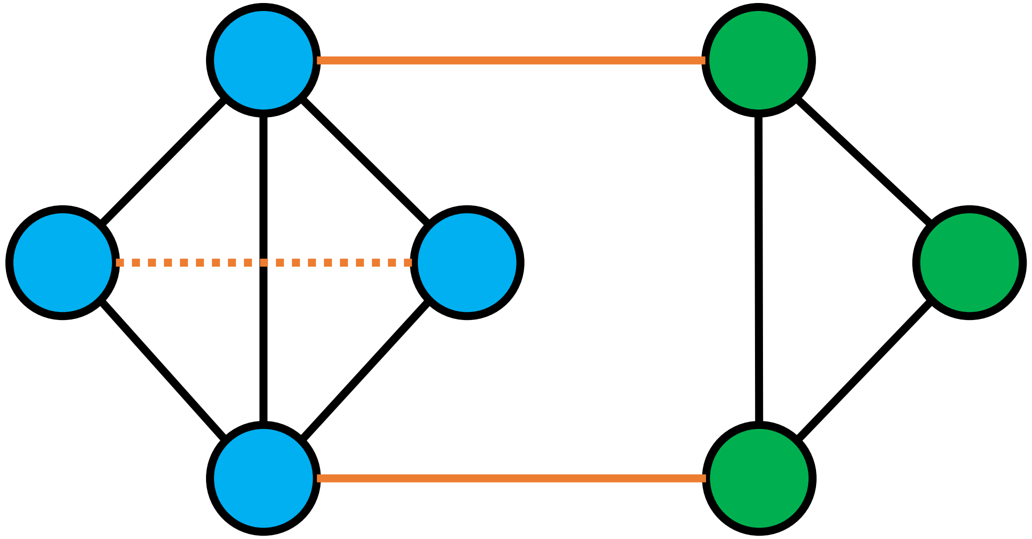
$$x_i = \{1, 2, 3, \dots, n\}$$

How to break the symmetries?



Option #1

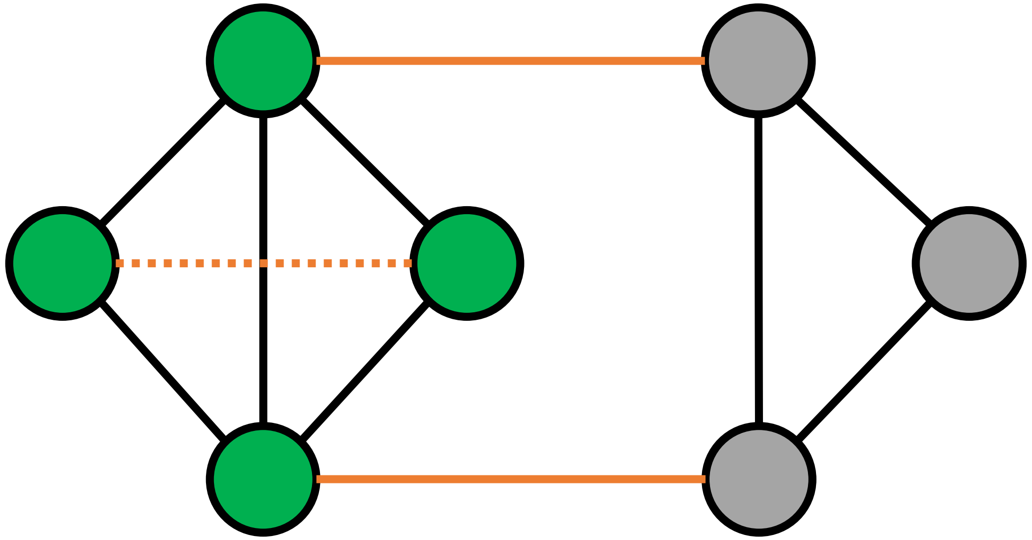
$x_i = \{1, 2, 3, \ldots, n\}$



Option #2

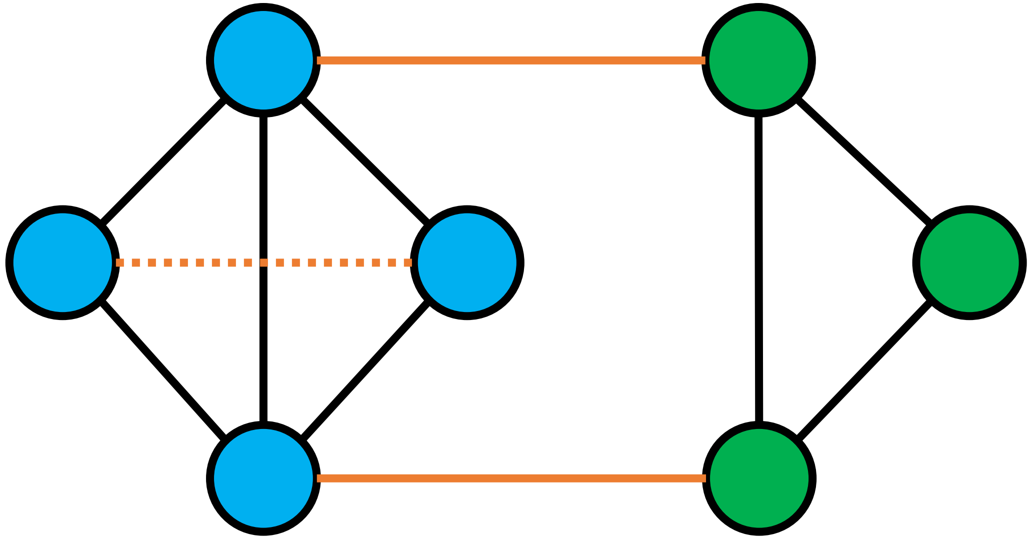
$y_{i,j} = \{0,1\}$

"are nodes i and j the same colour?"



Option #1

$$x_i = \{1, 2, 3, \ldots, n\}$$



Option #2

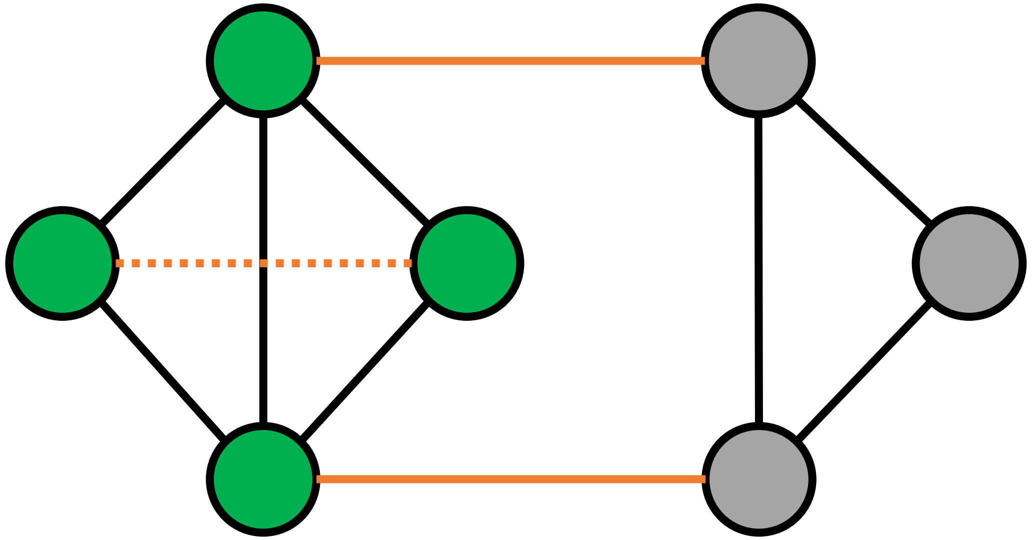
$$y_{i,j} = \{0,1\}$$

$$\forall i, j, k:$$

$$y_{i,j} \wedge y_{j,k} \rightarrow y_{i,k}$$

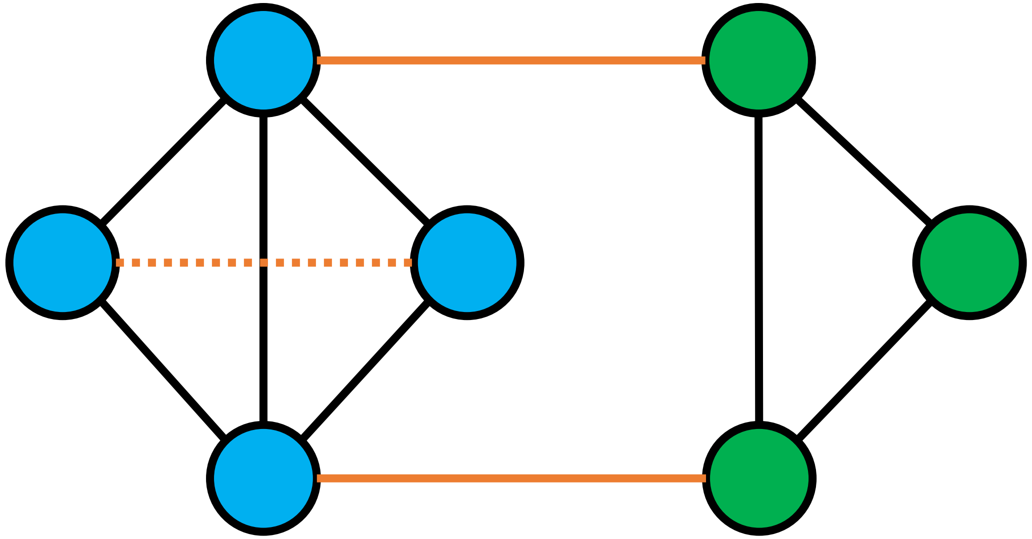
Must encode transitivity constraints to define the variables properly!

$O(n^3)$ constraints



Option #1

$$x_i = \{1, 2, 3, \ldots, n\}$$



Option #2

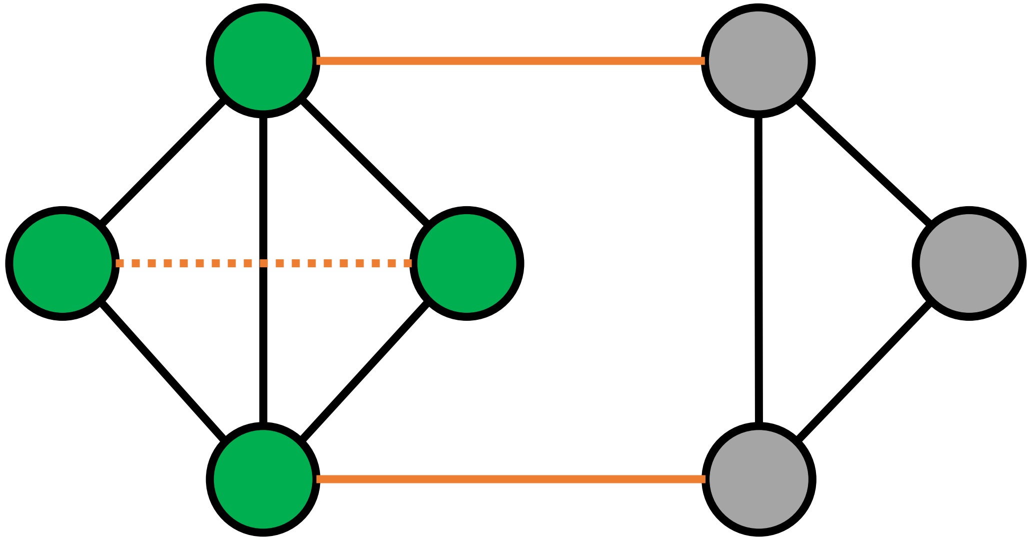
$$y_{i,j} = \{0,1\}$$

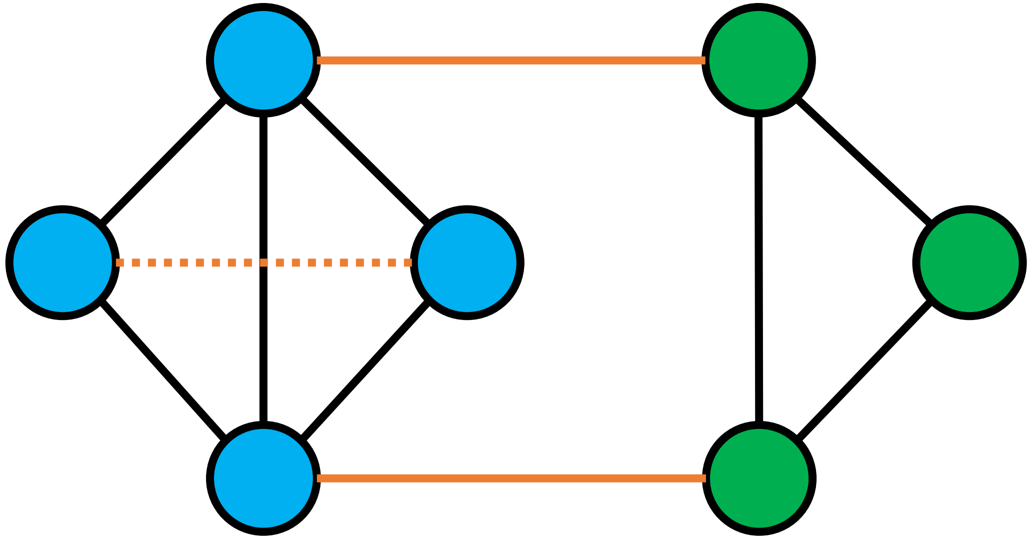
$$\forall i, j, k:$$

$$y_{i,j} \wedge y_{j,k} \rightarrow y_{i,k}$$

Must encode transitivity constraints
to define the variables properly!

$O(n^3)$
constraints





Option #1

$$x_i = \{1, 2, 3, \dots, n\}$$

"x_i is the colour of node i"

Option #2

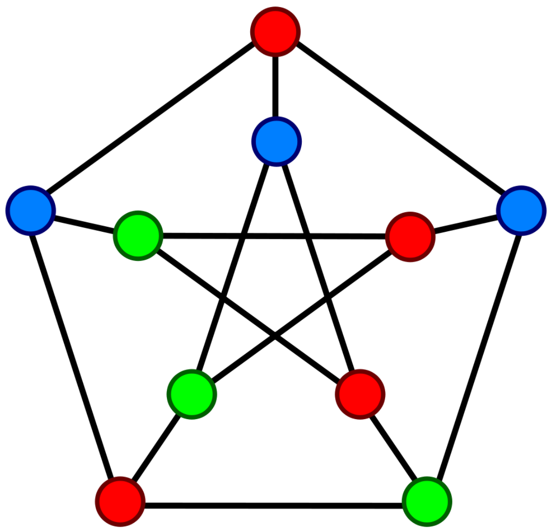
$$y_{i,j} = \{0,1\}$$

"are nodes i and j the same colour?"

Less constraints
but has symmetries

No symmetries
but many constraints

Graph Colouring

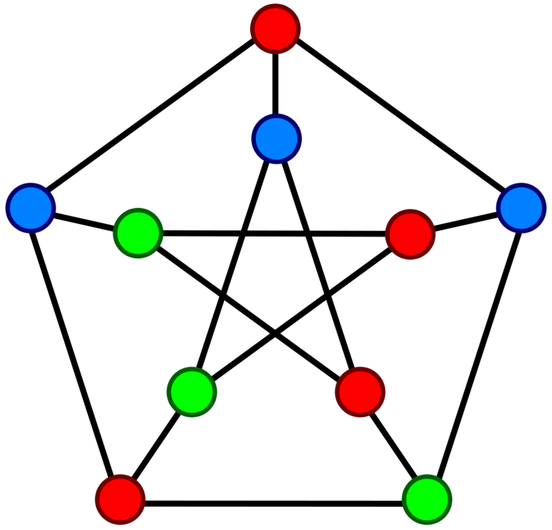


Where is the symmetry?

$$x_i \neq x_j$$

Image taken from Wikipedia

Graph Colouring

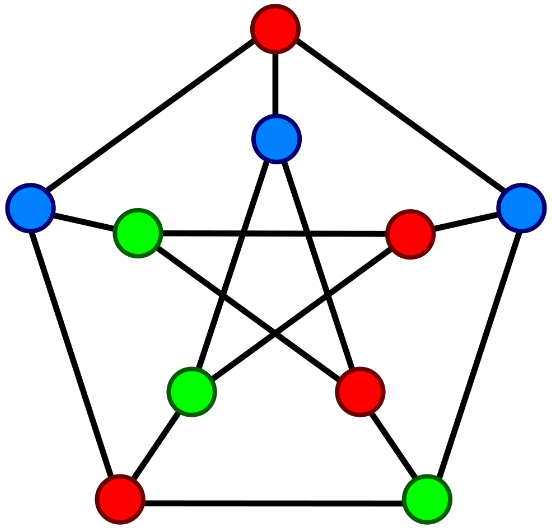


Colours can be renamed!

$$x_i \neq x_j$$

Image taken from Wikipedia

Graph Colouring



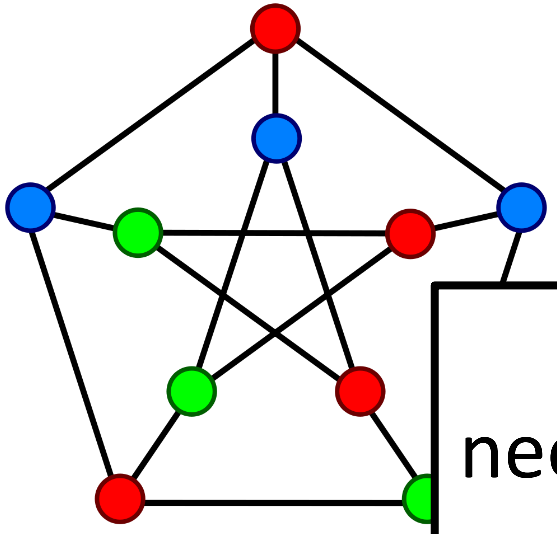
increases
search space!

Colours can be renamed!

Given a solution with k colours,
we can obtain $k!$ symmetric solutions

Image taken from Wikipedia

Graph Colouring



No easy workaround,
need algorithmic techniques
outside of this course!

Colours can be renamed!

Given a solution with k colours,
we can obtain $k!$ symmetric solutions

Image taken from Wikipedia

Symmetries

Symmetries generally undesirable
since they increase the search space

Symmetries can be removed by…

Symmetry breaking constraints

Different modelling

May be difficult!

Removing symmetries comes at a trade-off,
but usually a good idea!

Goal for today…

Search Tree

Exponential Growth

Modelling Patterns

Global Constraints

Symmetry breaking

Preprocessing and Relaxations

Emir Demirović

Algorithmics group | TU Delft

Algorithms for NP-Hard Problems (CSE2310 2023)

Lukina, Demirović, Yorke-Smith

Practical issues: Assignment 2

Hint

Permutation problems are weakly constrained combinatorial problems

Enumeration:

1. Find solution
2. Add constraint to forbid the solution
3. Repeat

Deadline Extension

March 20

Last time...

Search Tree

Exponential Growth → 10! is less than 0.000000001% of 20!

Modelling Patterns

Global Constraints

Symmetry breaking

Goal for today…

Preprocessing

Relaxations

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Preprocessing

“simplify the problem (with expensive/special reasoning) before solving”

Remove duplicate constraints

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Remove duplicate constraints

Remove subsumed constraints

$$x_1 + x_2 \geq 1 \qquad x_1 + x_2 + x_3 \geq 1$$

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Remove duplicate constraints

Remove subsumed constraints

$x_1 + x_2 \geq 1$

$\cancel{x_1 + x_2 + x_3 \geq 1}$

can remove!

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Remove duplicate constraints

Remove subsumed constraints

Reason over combinations of constraints

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Remove duplicate constraints

Remove subsumed constraints

Reason over combinations of constraints

$$\left.\begin{aligned} x_1 + x_2 + x_3 &\geq 1 \\ x_1 + x_2 - x_3 &\geq 0 \end{aligned}\right\}$$

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Remove duplicate constraints

Remove subsumed constraints

Reason over combinations of constraints

$$\left.\begin{array}{l} x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 - x_3 \geq 0 \end{array}\right\} \xrightarrow[\text{Add up constraints}]{} 2x_1 + 2x_2 \geq 1$$

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Remove duplicate constraints

Remove subsumed constraints

Reason over combinations of constraints

$$\left.\begin{array}{l} x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 - x_3 \geq 0 \end{array}\right\} \xrightarrow[\text{Add up constraints}]{} 2x_1 + 2x_2 \geq 1 \xrightarrow[x_i \in \mathbb{N}_0]{\text{"saturation"}} x_1 + x_2 \geq 1$$

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Remove duplicate constraints

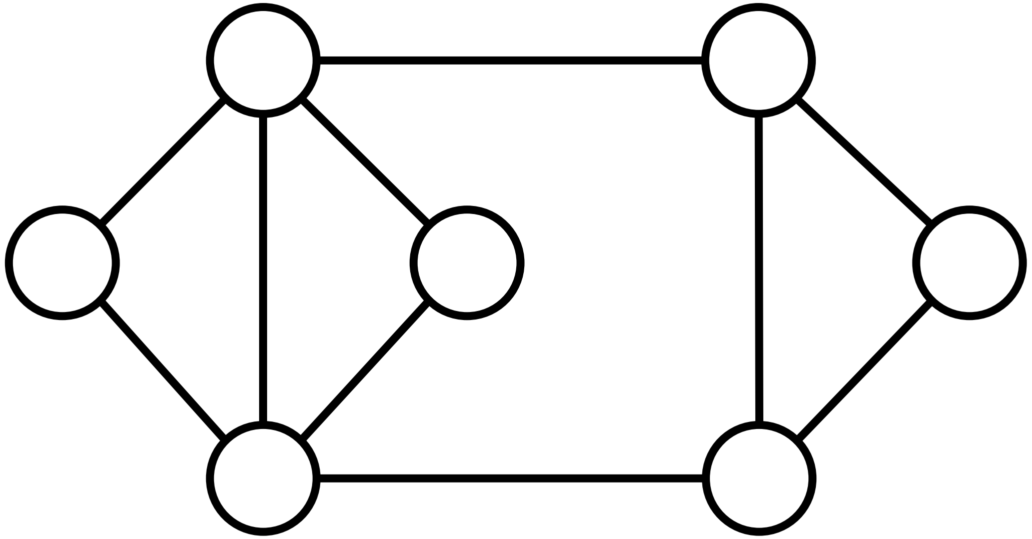
Remove subsumed constraints

Reason over combinations of constraints

$$\left.\begin{array}{l} \cancel{x_1 + x_2 + x_3 \geq 1} \\ \cancel{x_1 + x_2 - x_3 \geq 0} \end{array}\right\} \longrightarrow 2x_1 + 2x_2 \geq 1 \longrightarrow x_1 + x_2 \geq 1$$

Derived constraint implied, but the original constraints now subsumed!

Preprocessing: Decomposition

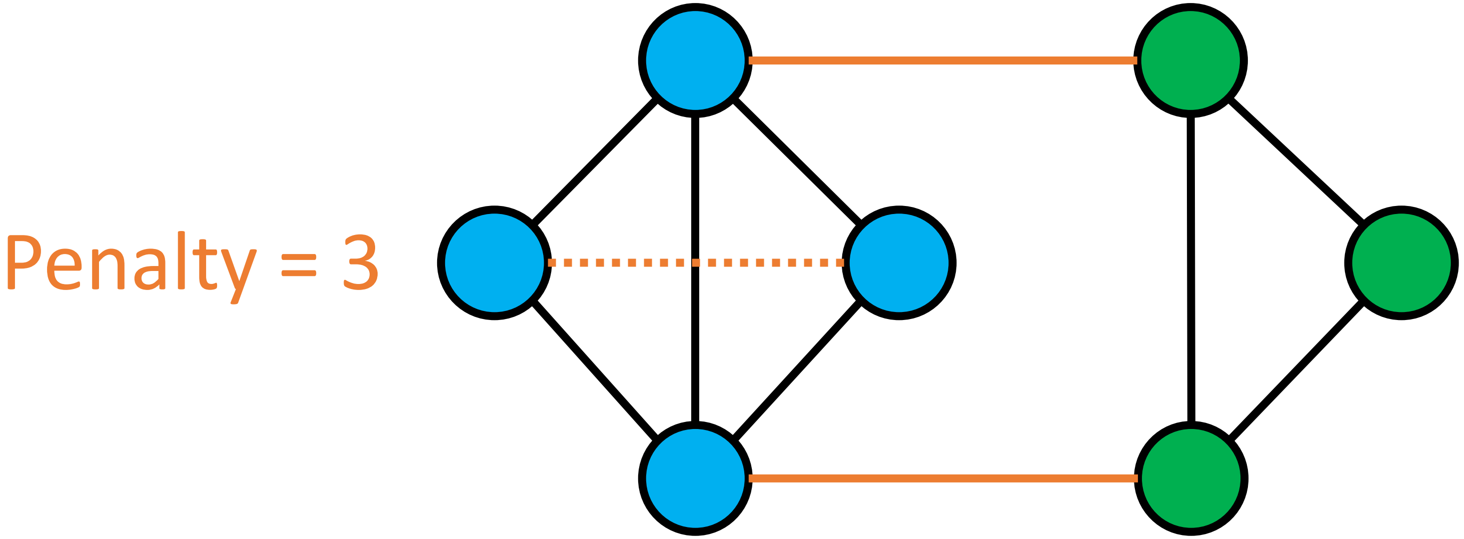


Colour nodes

Connected nodes preferably in same colour

Disconnected nodes preferably in different colours

Preprocessing: Decomposition

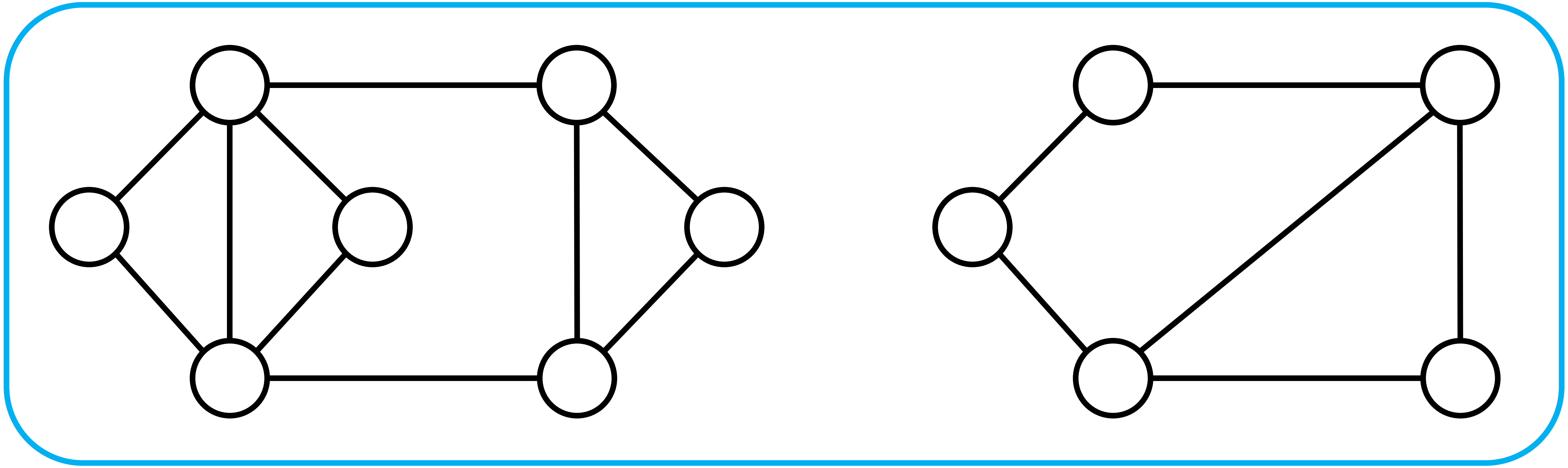


Colour nodes

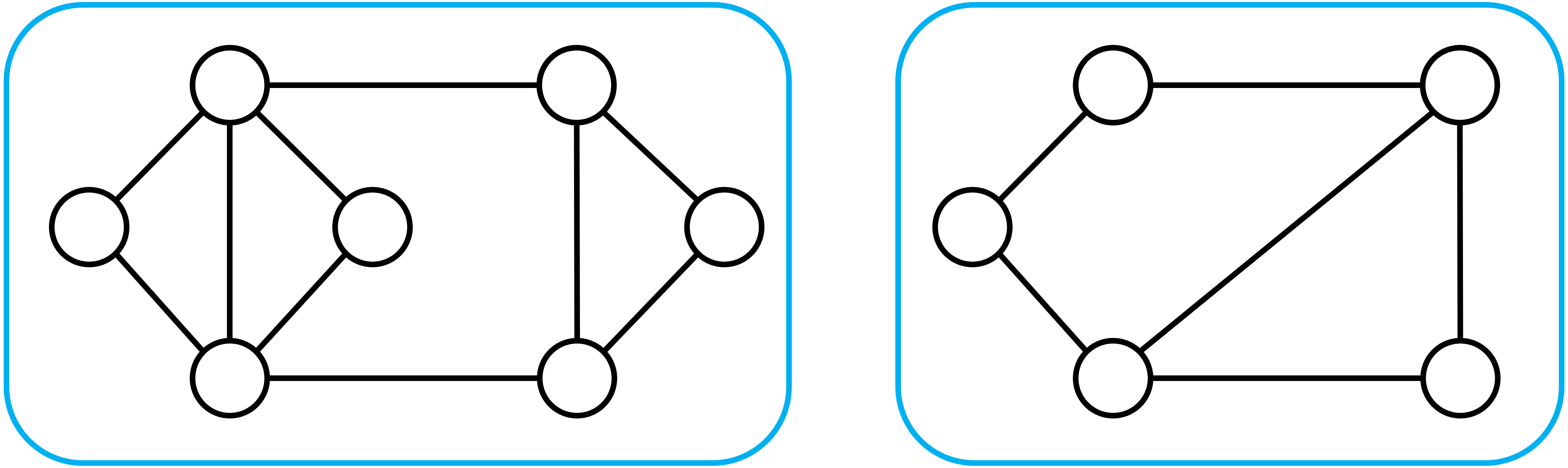
Connected nodes preferably in same colour

Disconnected nodes preferably in different colours

Preprocessing: Decomposition



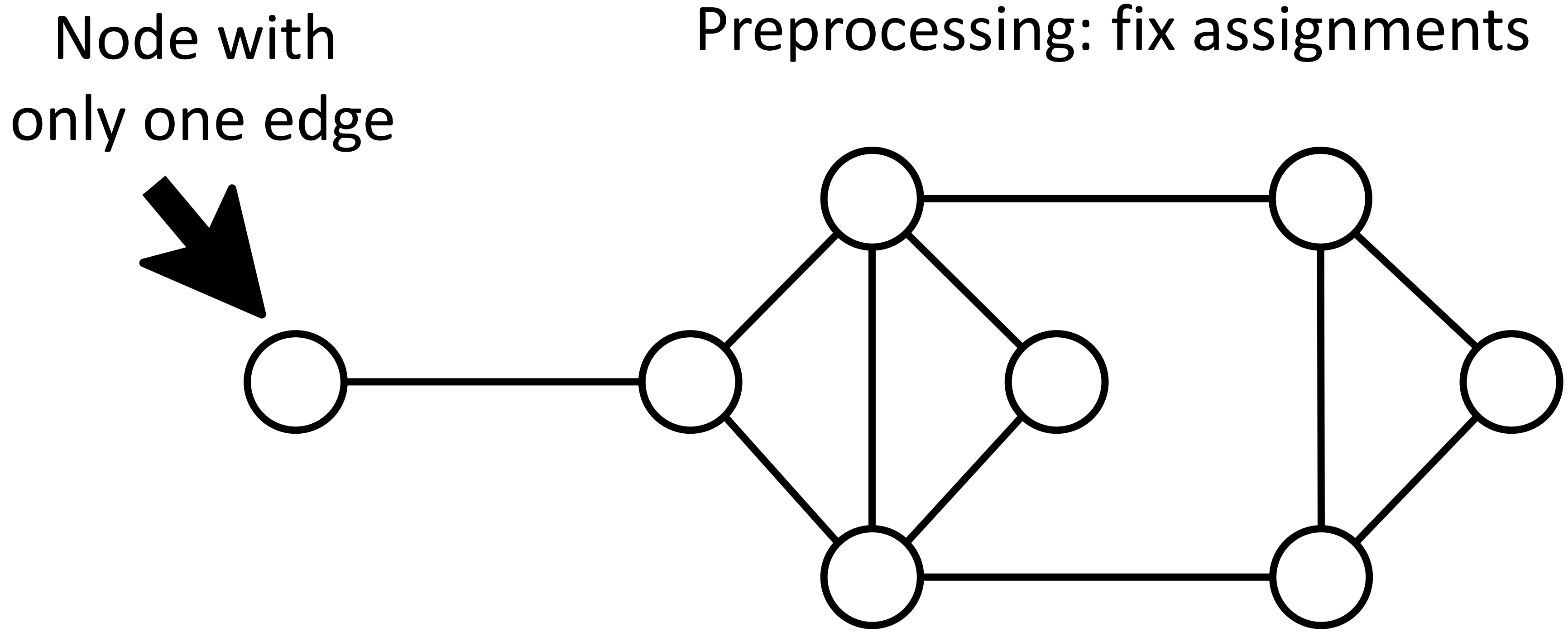
Preprocessing: Decomposition



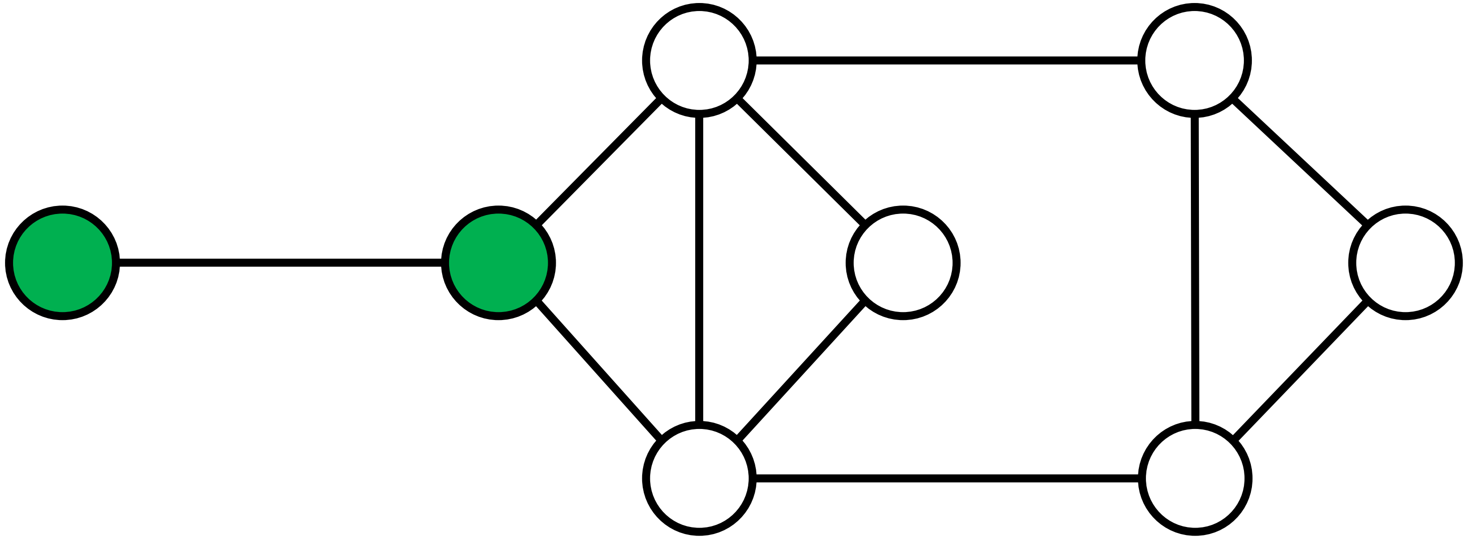
Compute strongly connected components in $O(num\_nodes)$

Each component can be solved independently!

Preprocessing: fix assignments



Preprocessing: fix assignments



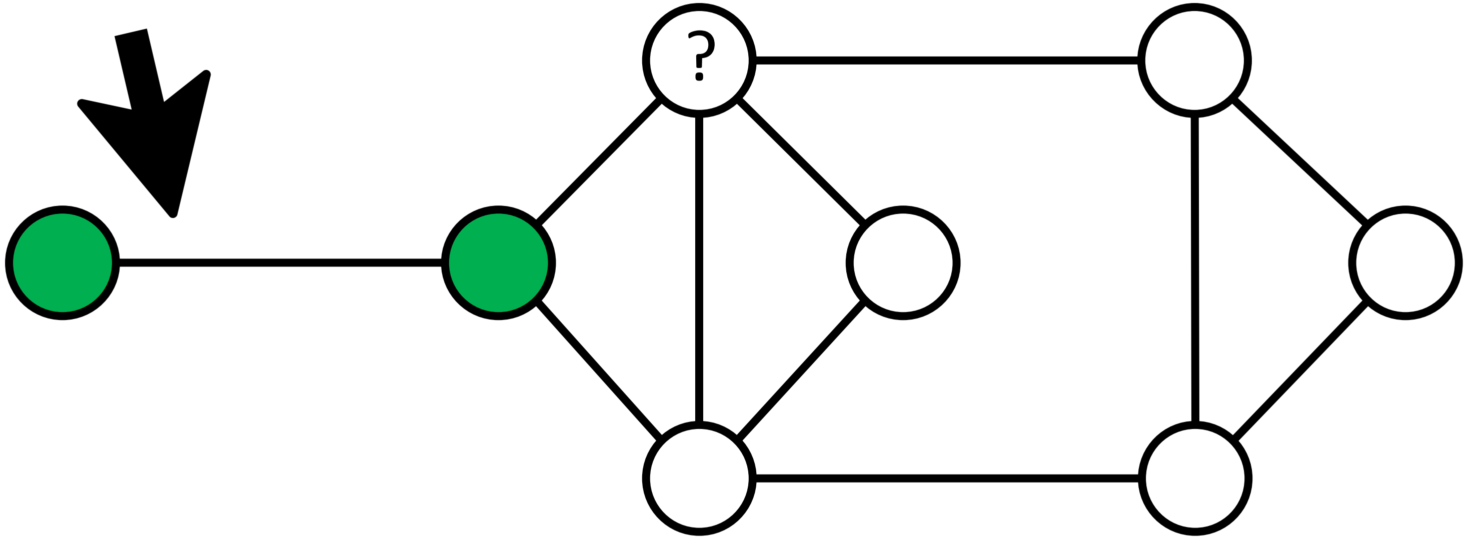
Case reasoning:

1. Node will be assigned colour of neighbour

Disclaimer: small mix-up in the live lecture! Fixed here

Preprocessing: fix assignments

No penalty

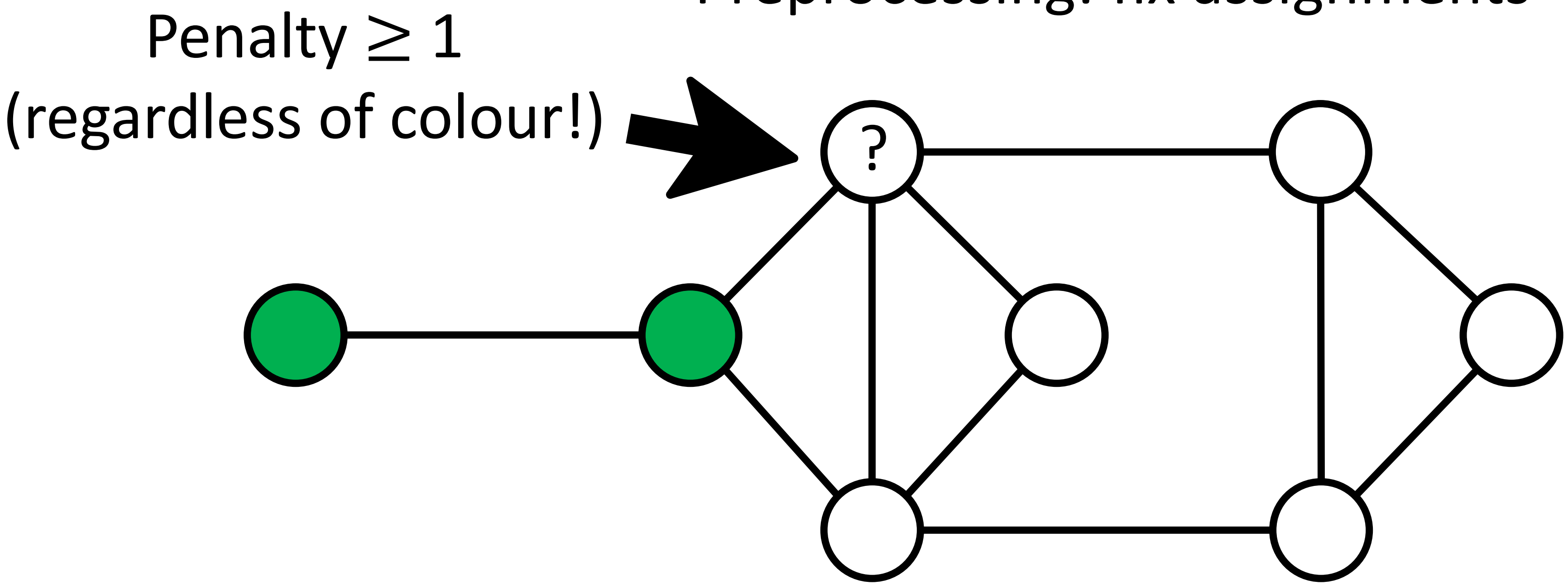


Case reasoning:

1. Node will be assigned colour of neighbour

Disclaimer: small mix-up in the live lecture! Fixed here

Preprocessing: fix assignments

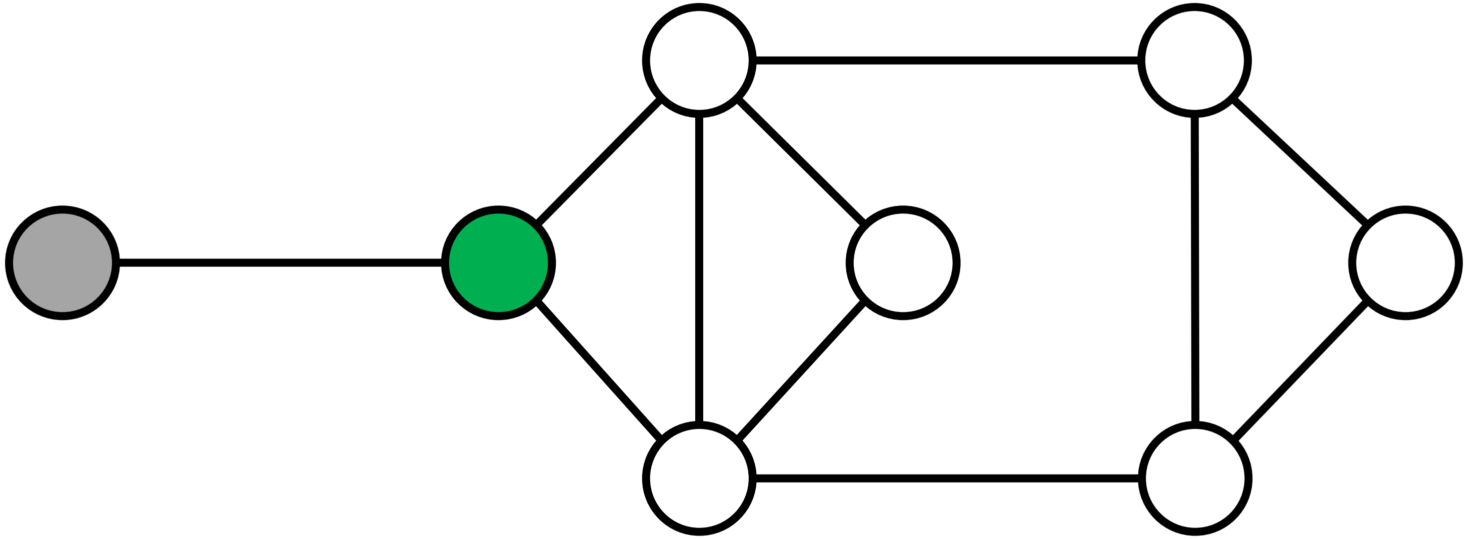


Case reasoning:

1. Node will be assigned colour of neighbour

Disclaimer: small mix-up in the live lecture! Fixed here

Preprocessing: fix assignments



Case reasoning:

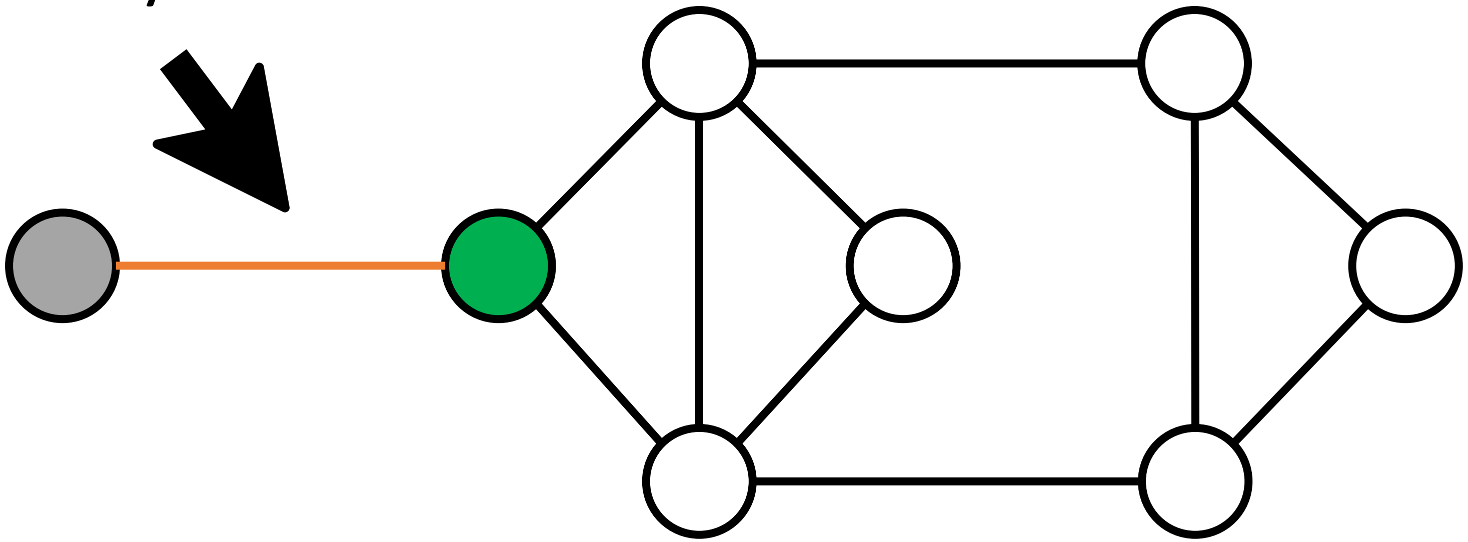
1. Node will be assigned colour of neighbour Penalty ≥ 1

2. Node will no be assigned colour of neighbour

Disclaimer: small mix-up in the live lecture! Fixed here

Preprocessing: fix assignments

Penalty $= 1$



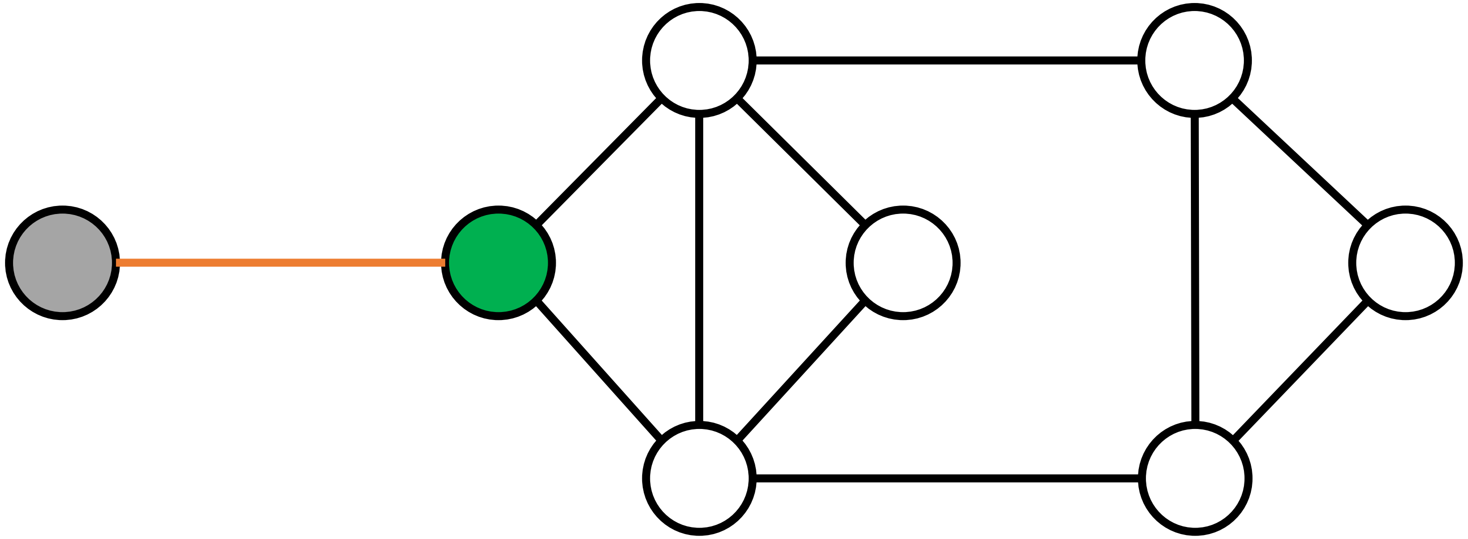
Case reasoning:

1. Node will be assigned colour of neighbour   Penalty ≥ 1

2. Node will no be assigned colour of neighbour

Disclaimer: small mix-up in the live lecture! Fixed here

Preprocessing: fix assignments



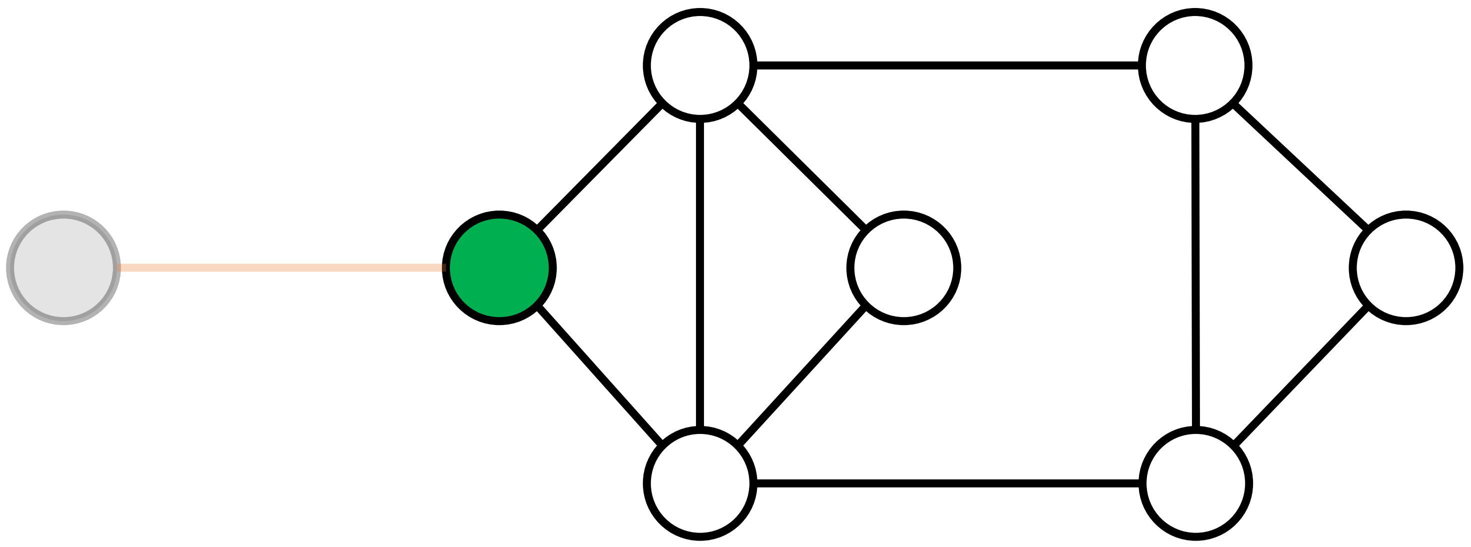
Case reasoning:

1. Node will be assigned colour of neighbour — Penalty ≥ 1

2. Node will no be assigned colour of neighbour — Penalty $= 1$

Disclaimer: small mix-up in the live lecture! Fixed here

Preprocessing: fix assignments



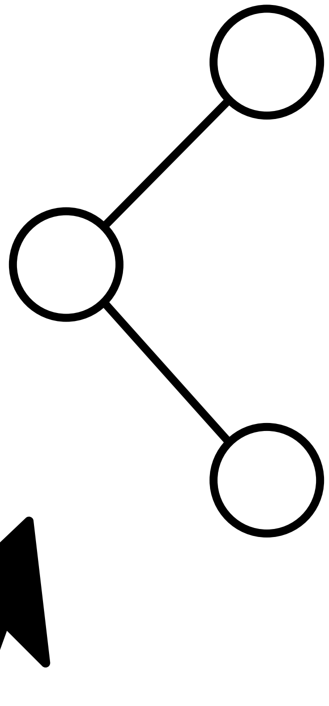
Case reasoning:

1. Node will be assigned colour of neighbour Penalty ≥ 1

2. Node will no be assigned colour of neighbour Penalty $= 1$

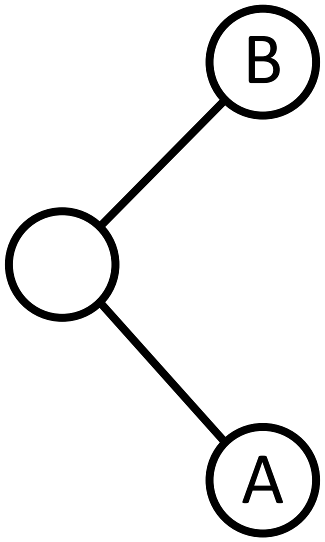
Conclusion: Remove node in preprocessing, assign colour different than neighbour

Preprocessing: additional constraints



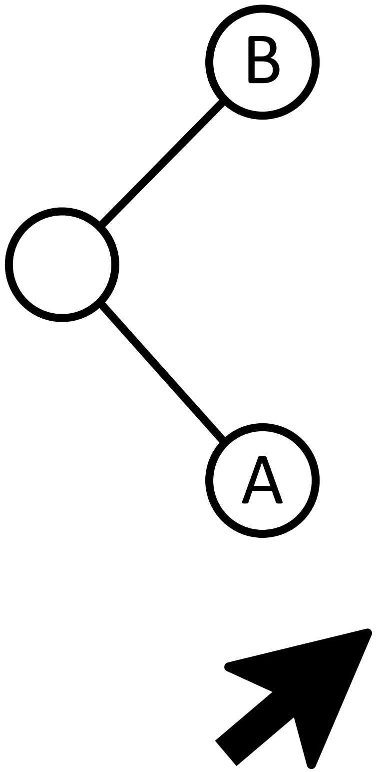
subgraph

Preprocessing: additional constraints



Every colouring leads to
at least one violation!

Preprocessing: additional constraints

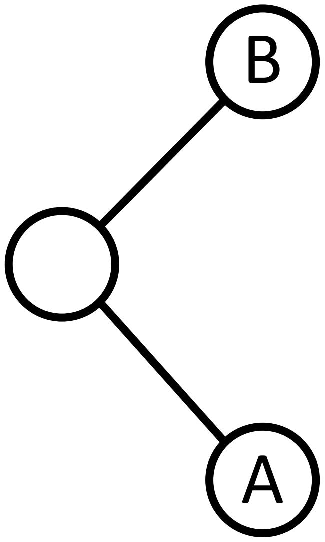


Every colouring leads to
at least one violation!

Add constraint:

“node A and node B have different colours”

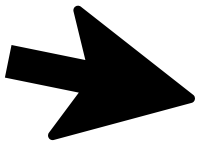
Preprocessing: additional constraints



Every colouring leads to
at least one violation!

Add constraint:

"node A and node B have different colours"



Removes solutions, but at least
one optimal solution remains!

More complex rules in TU Delft bachelor thesis
Rhebergen Damiaan, "Cluster Editing with Diamond-free Vertices". 2021

Preprocessing: Stronger inference at the root level

Look Ahead

“Limited breadth-first search”

Inference from tentative assignments

Preprocessing: Stronger inference at the root level

Look Ahead

"Limited breadth-first search"

For each $v \in domain(x_i)$:

Preprocessing: Stronger inference at the root level

Look Ahead

"Limited breadth-first search"

For each $v \in domain(x_i)$:

Preprocessing: Stronger inference at the root level

Look Ahead

"Limited breadth-first search"

For each $v \in domain(x_i)$:

Assign $x_i = v$

Propagate

Unassigned x_i

Preprocessing: Stronger inference at the root level

Look Ahead

"Limited breadth-first search"

For each $v \in domain(x_i)$:

  Assign $x_i = v$

  Propagate

  Unassigned x_i

  If conflict detected before:

    Remove v from domain

Preprocessing: Stronger inference at the root level

Look Ahead

"Limited breadth-first search"

For each $v \in domain(x_i)$:

Assign $x_i = v$

Propagate

Unassigned x_i

If conflict detected before:

Remove v from domain

If each time v' was removed from domain of x_j:

Remove v' from x_j

Preprocessing: Stronger inference at the root level

Look Ahead

"Limited breadth-first search"

For each $v \in domain(x_i)$:

Assign $x_i = v$

Propagate

Unassigned x_i

If conflict detected before:

Remove v from domain

If each time v' was removed from domain of x_j:

Remove v' from x_j

Repeat for each variable x_i

Preprocessing: Stronger inference at the root level

Look Ahead

"Limited breadth-first search"

Not fully committing to depth-first search

For each $v \in domain(x_i)$:

Assign $x_i = v$

Propagate

Unassigned x_i

If conflict detected before:

Remove v from domain

If each time v' was removed from domain of x_j:

Remove v' from x_j

Look Ahead

$x_i \in \{0,1\}$

c_1: $x_1 + x_2 - x_3 \geq 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

… …

… …

… …

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

$x_1 = 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

… …

… …

… …

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

$x_1 = 0$

No propagations!

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

… …

… …

… …

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

$x_1 = 0$

c_2: $-x_2 + x_3 \geq 0$

No propagations!

c_3: $x_2 + x_3 \geq 1$

$x_1 = 1$

c_4: $-x_3 + x_4 \geq 0$

… …

… …

… …

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

… …

… …

… …

$x_1 = 0$

No propagations!

$x_1 = 1$

No propagations!

Did not gain anything
from look ahead!

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

$x_2 = 0$

Look ahead assignment

… …

… …

… …

$x_i \in \{0,1\}$

Look Ahead

$c_1\colon\ x_1 + x_2 - x_3 \geq 0$

$x_2 = 0$

$c_2\colon -x_2 + x_3 \geq 0$

$c_3\colon x_2 + x_3 \geq 1$

$c_4\colon -x_3 + x_4 \geq 0$

… …

… …

… …

$x_i \in \{0,1\}$

Look Ahead

$c_1\colon x_1 + x_2 - x_3 \geq 0$

$x_2 = 0$

$c_3\colon x_3 = 1$

$c_2\colon -x_2 + x_3 \geq 0$

$c_3\colon x_2 + x_3 \geq 1$

$c_4\colon -x_3 + x_4 \geq 0$

… …

… …

… …

$x_i \in \{0,1\}$

Look Ahead

$c_1\!: x_1 + x_2 - x_3 \geq 0$

$c_2\!: -x_2 + x_3 \geq 0$

$c_3\!: x_2 + x_3 \geq 1$

$c_4\!: -x_3 + x_4 \geq 0$

… …

… …

… …

$x_2 = 0$

$c_3\!: x_3 = 1$

$c_1\!: x_1 = 1$

$c_4\!: x_4 = 1$

$x_i \in \{0,1\}$

Look Ahead

$c_1:\ x_1 + x_2 - x_3 \geq 0$

$c_2: -x_2 + x_3 \geq 0$

$c_3: x_2 + x_3 \geq 1$

$c_4: -x_3 + x_4 \geq 0$

… …

… …

… …

$x_2 = 0$

$c_3: x_3 = 1$

$c_1: x_1 = 1$

$c_4: x_4 = 1$

$x_2 = 1$

Look ahead assignment

$x_i \in \{0,1\}$

Look Ahead

$c_1\text{: } x_1 + x_2 - x_3 \geq 0$

$c_2\text{: } -x_2 + x_3 \geq 0$

$c_3\text{: } x_2 + x_3 \geq 1$

$c_4\text{: } -x_3 + x_4 \geq 0$

… …

… …

… …

$x_2 = 0$

$c_3\text{: } x_3 = 1$

$c_1\text{: } x_1 = 1$

$c_4\text{: } x_4 = 1$

$x_2 = 1$

$c_2\text{: } x_3 = 1$

$c_4\text{: } x_4 = 1$

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

… …

… …

… …

$x_2 = 0$

$c_3: x_3 = 1$

$c_1: x_1 = 1$

$c_4: x_4 = 1$

$x_2 = 1$

$c_2: x_3 = 1$

$c_4: x_4 = 1$

What can we conclude?

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

… …

… …

… …

$x_2 = 0$

$c_3: x_3 = 1$

$c_1: x_1 = 1$

$c_4: x_4 = 1$

$x_2 = 1$

$c_2: x_3 = 1$

$c_4: x_4 = 1$

Regardless of x_2,
$x_3 = 1$ and $x_4 = 1$

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

… …

… …

… …

$x_3 = 1$
$x_4 = 1$

Assignments before search started!

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

c_5: $x_5 + x_6 - x_3 \geq 1$

c_6: $x_5 - x_6 - x_4 \geq -1$

… …

$x_3 = 1$

$x_4 = 1$

$x_5 = 0$

Look ahead assignment

$x_i \in \{0,1\}$

Look Ahead

$c_1 : x_1 + x_2 - x_3 \geq 0$

$x_3 = 1$

$c_2 : -x_2 + x_3 \geq 0$

$x_4 = 1$

$c_3 : x_2 + x_3 \geq 1$

$x_5 = 0$

$c_4 : -x_3 + x_4 \geq 0$

$\boxed{c_5 : x_5 + x_6 - x_3 \geq 1} \rightarrow -1 + x_6 \geq 1$

$c_6 : x_5 - x_6 - x_4 \geq -1$

… …

$x_i \in \{0,1\}$

Look Ahead

$c_1: x_1 + x_2 - x_3 \geq 0$

$c_2: -x_2 + x_3 \geq 0$

$c_3: x_2 + x_3 \geq 1$

$c_4: -x_3 + x_4 \geq 0$

$\boxed{c_5: x_5 + x_6 - x_3 \geq 1}$ $\rightarrow$ $-1 + x_6 \geq 1$ $\rightarrow$ **Conflict!**

$c_6: x_5 - x_6 - x_4 \geq -1$

… …

$x_3 = 1$

$x_4 = 1$

$x_5 = 0$

$x_i \in \{0,1\}$

Look Ahead

$c_1: x_1 + x_2 - x_3 \geq 0$

$c_2: -x_2 + x_3 \geq 0$

$c_3: x_2 + x_3 \geq 1$

$c_4: -x_3 + x_4 \geq 0$

$$x_3 = 1$$
$$x_4 = 1$$
$$x_5 = 0$$

$\boxed{c_5: x_5 + x_6 - x_3 \geq 1}$ → $-1 + x_6 \geq 1$ → **Conflict!** → $x_5 \neq 0$

$c_6: x_5 - x_6 - x_4 \geq -1$

… …

$x_i \in \{0,1\}$

Look Ahead

$c_1\colon x_1 + x_2 - x_3 \geq 0$

$c_2\colon -x_2 + x_3 \geq 0$

$c_3\colon x_2 + x_3 \geq 1$

$c_4\colon -x_3 + x_4 \geq 0$

$c_5\colon x_5 + x_6 - x_3 \geq 1$

$c_6\colon x_5 - x_6 - x_4 \geq -1$

… …

$x_3 = 1$

$x_4 = 1$

$x_5 = 1$

$c_5\colon x_6 = 1$

Assignments
before search started!

$x_i \in \{0,1\}$

Look Ahead

c_1: $x_1 + x_2 - x_3 \geq 0$

c_2: $-x_2 + x_3 \geq 0$

c_3: $x_2 + x_3 \geq 1$

c_4: $-x_3 + x_4 \geq 0$

c_5: $x_5 + x_6 - x_3 \geq 1$

c_6: $x_5 - x_6 - x_4 \geq -1$

… …

$x_3 = 1$

$x_4 = 1$

$x_5 = 1$

c_5: $x_6 = 1$

Look-ahead provides more propagation,
but computationally expensive!

…usually something to consider
in preprocessing and/or for a small subset of variables

Preprocessing

"simplify the problem (with expensive/special reasoning) before solving"

Remove duplicate constraints

Remove subsumed constraints

Reason over combinations of constraints

Add additional constraints to narrow the search

Use stronger reasoning before starting search

Look-Ahead

Relaxation

“simplified version of the original problem”

Relaxation

"simplified version of the original problem"

NP-hard problems difficult to solve

Simplified problem can still provide valuable information

Relaxation

"simplified version of the original problem"

$$\max \sum_j w_j \cdot x_j$$

$$\forall i: \sum_j c_{ij} \cdot x_j \geq k_j$$

$$x_i \in \{0,1\}$$

Relaxation

"simplified version of the original problem"

$$\max \sum_j w_j \cdot x_j$$

$$\forall i: \sum_j c_{ij} \cdot x_j \geq k_j$$

$$x_i \in \{0,1\}$$

difficult to solve
because of "integrality constraints"

Relaxation

"simplified version of the original problem"

$$\max \sum_j w_j \cdot x_j$$

$$\forall i: \sum_j c_{ij} \cdot x_j \geq k_j$$

$$x_i \in \{0,1\}$$

difficult to solve
(Integer Program)

$$\max \sum_j w_j \cdot x_j$$

$$\forall i: \sum_j c_{ij} \cdot x_j \geq k_j$$

$$x_i \in [0, 1]$$

easy to solve
(Linear Program)

Relaxation

"simplified version of the original problem"

Relaxation of the problem

$$\max \sum_j w_j \cdot x_j$$

$$\forall i\colon \sum_j c_{ij} \cdot x_j \geq k_j$$

$$x_i \in \{0,1\}$$

$$\max \sum_j w_j \cdot x_j$$

$$\forall i\colon \sum_j c_{ij} \cdot x_j \geq k_j$$

$$x_i \in [0,1]$$

difficult to solve
(Integer Program)

easy to solve
(Linear Program)

Relaxation

"simplified version of the original problem"

$$\max \sum_j w_j \cdot x_j$$

$$\forall i\colon \sum_j c_{ij} \cdot x_j \geq k_j$$

$$x_i \in \{0,1\}$$

Relaxation of the problem

$$\boxed{\begin{gathered}\max \sum_j w_j \cdot x_j \\ \forall i\colon \sum_j c_{ij} \cdot x_j \geq k_j \\ x_i \in [0,1]\end{gathered}}$$

Why are relaxations useful?

difficult to solve
(Integer Program)

easy to solve
(Linear Program)

Relaxation

"simplified version of the original problem"

Solving the relaxation provides…

A bound on the optimal objective function value

Adding more constraints can only make the objective worse!

Relaxation

“simplified version of the original problem”

Solving the relaxation provides…

A bound on the optimal objective function value

The optimal solution to the original problem if we are lucky

if the relaxed solution
also satisfies constraints of the original problem

Relaxation

"simplified version of the original problem"

Solving the relaxation provides…

A bound on the optimal objective function value

The optimal solution to the original problem if we are lucky

could think of propagation as reasoning over relaxations!
each individual constraint is a relaxation of the original problem

Relaxation

"simplified version of the original problem"

$$\max \sum_j w_j \cdot x_j$$

$$\forall i: \sum_j c_{ij} \cdot x_j \geq k_j$$

$$x_i \in \{0,1\}$$

$$\leq$$

Relaxation of the problem

$$\boxed{\begin{gathered} \max \sum_j w_j \cdot x_j \\ \forall i: \sum_j c_{ij} \cdot x_j \geq k_j \\ x_i \in [0,1] \end{gathered}}$$

difficult to solve
(Integer Program)

easy to solve
(Linear Program)

Constrained Shortest Path Problem

Given a graph

Find the shortest path between node A and node G

Using exactly k edges

NP-hard!

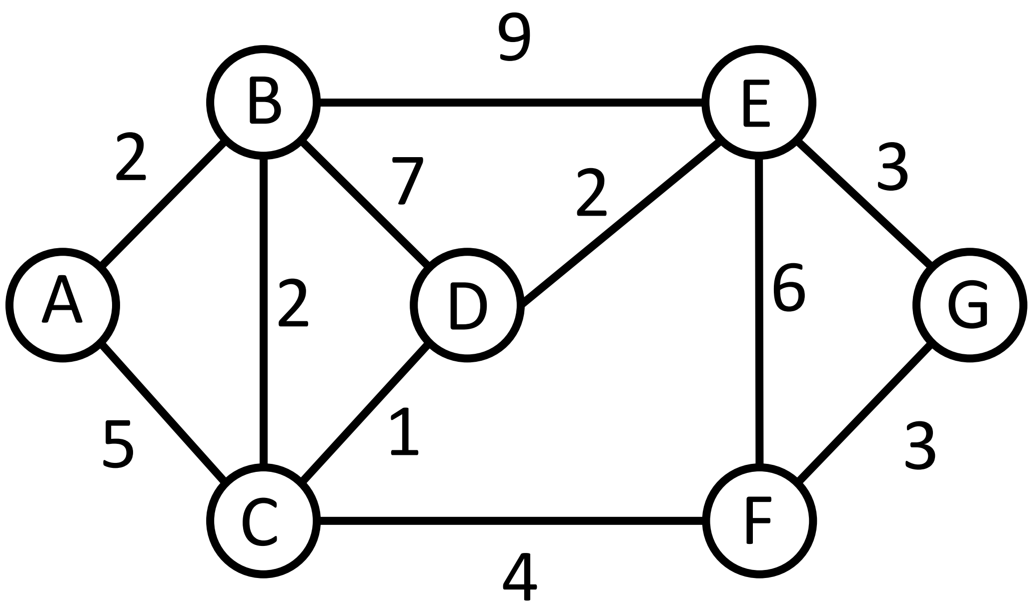
↓

See last slides for proof!

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges

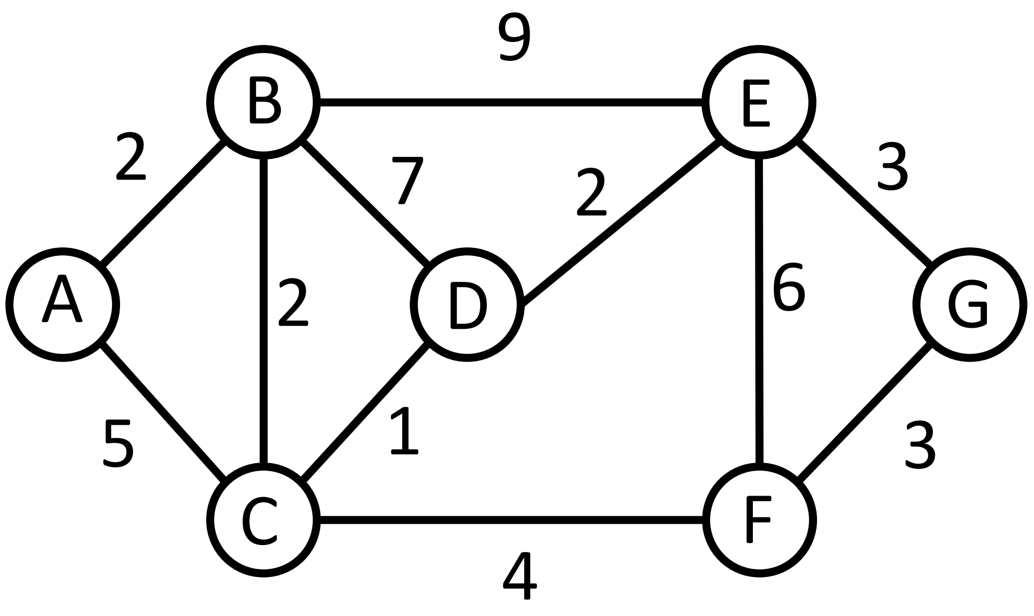


Constrained Shortest Path Problem

Compute shortest path between node A and node G

~~Use exactly four edges~~

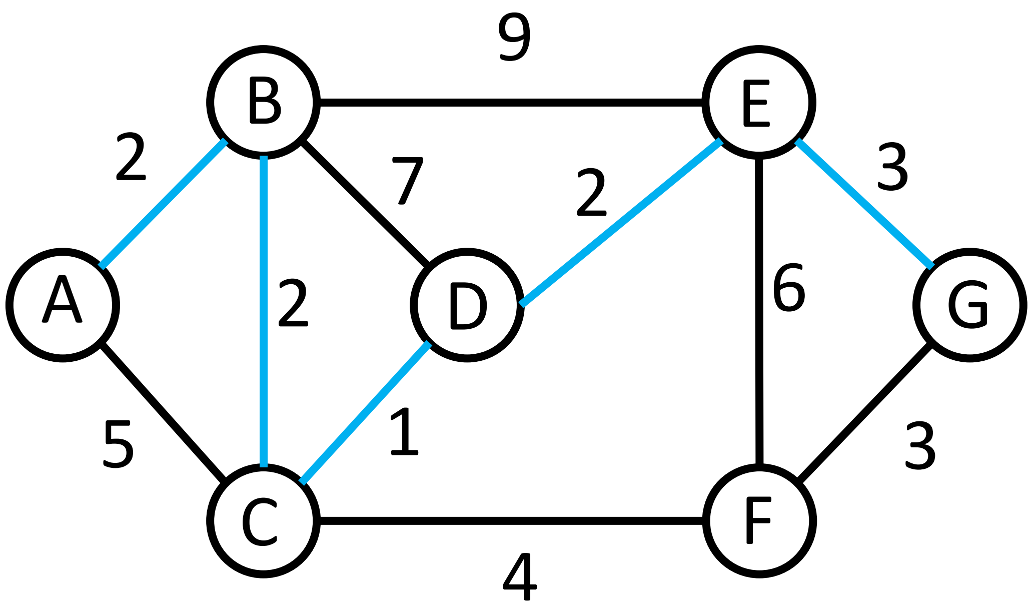
Relaxation!



Constrained Shortest Path Problem

Compute shortest path between node A and node G

~~Use exactly four edges~~



Relaxed cost = 10

Bound!

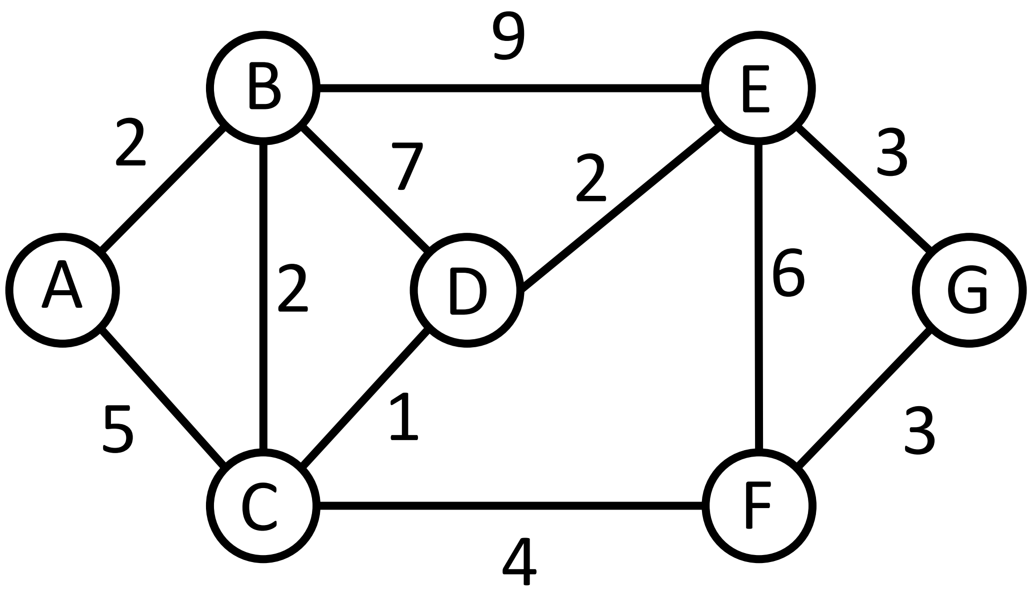
Relaxed solution infeasible

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges

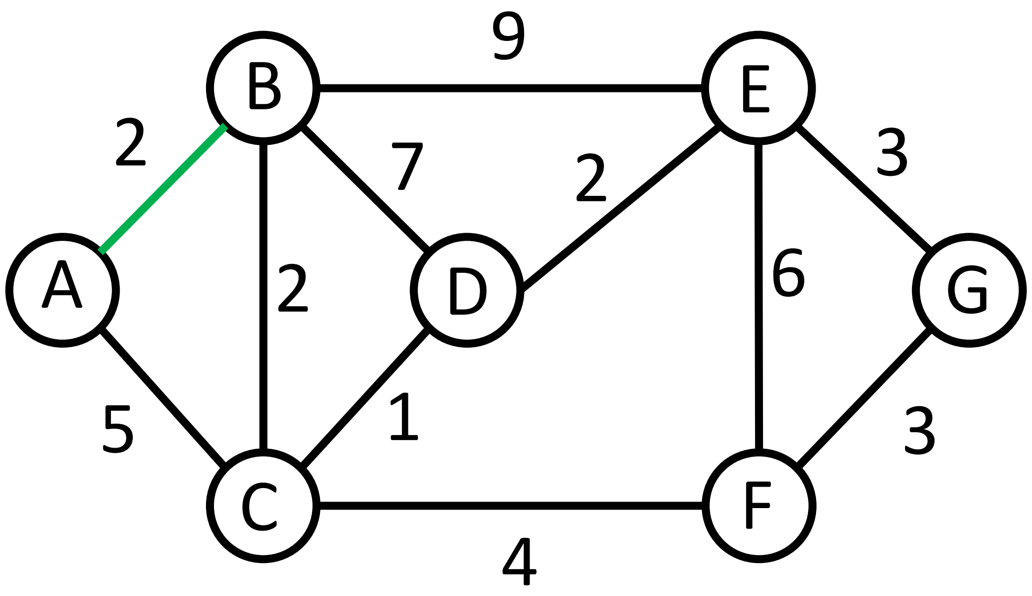
Cost	Relax.	Best Sol.
0	10	∞



Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges

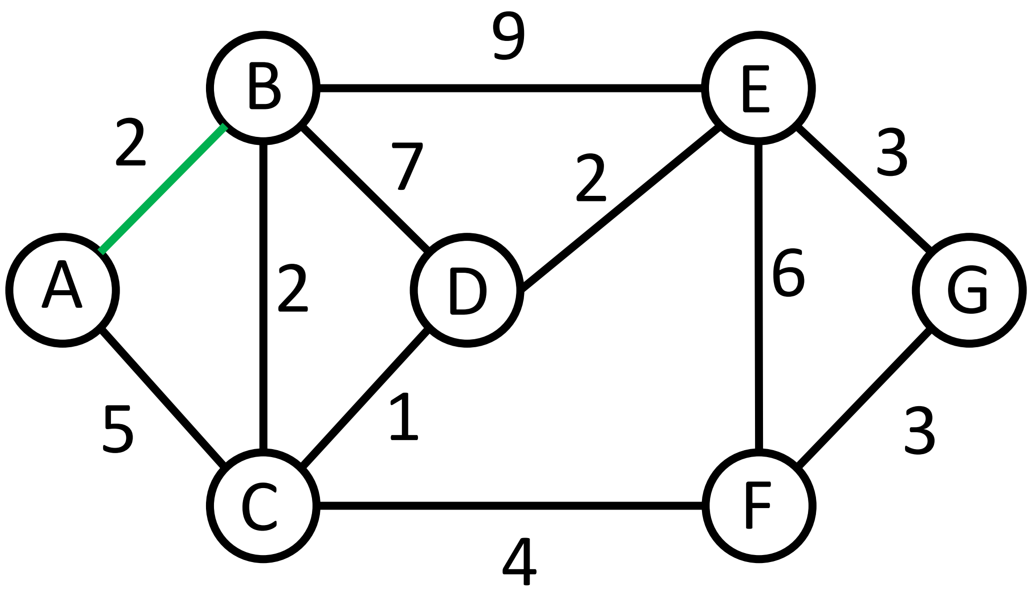


	Cost	Relax.	Best Sol.
	0	10	∞
$X_{A,B} = 1$	2	?	

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



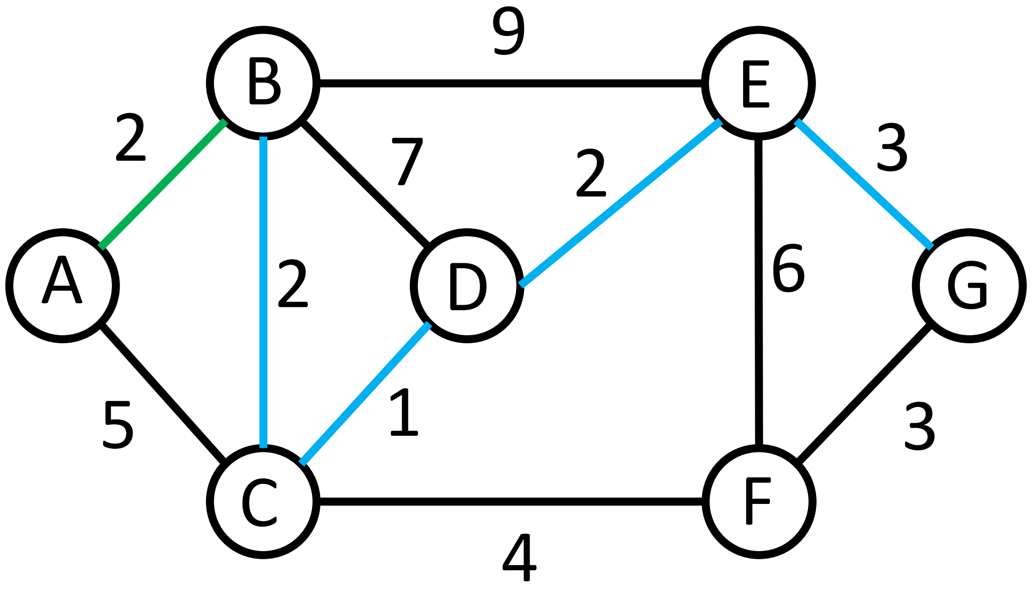
	Cost	Relax.	Best Sol.
	0	10	∞
$X_{A,B} = 1$	2	?	

Compute relaxation...

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



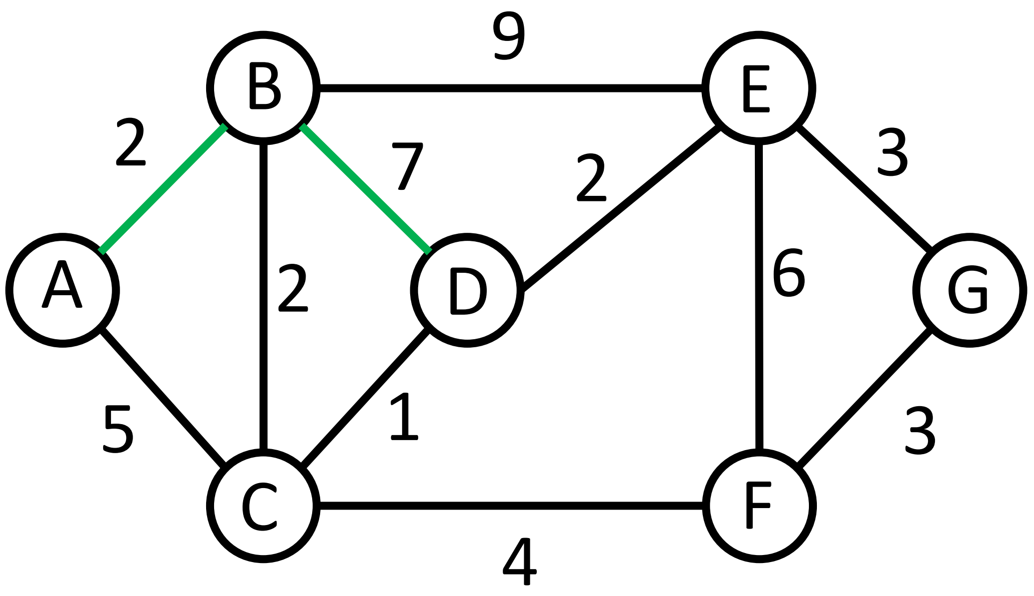
	Cost	Relax.	Best Sol.
	0	10	∞
$X_{A,B} = 1$	2	10	

Relaxed solution infeasible

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges

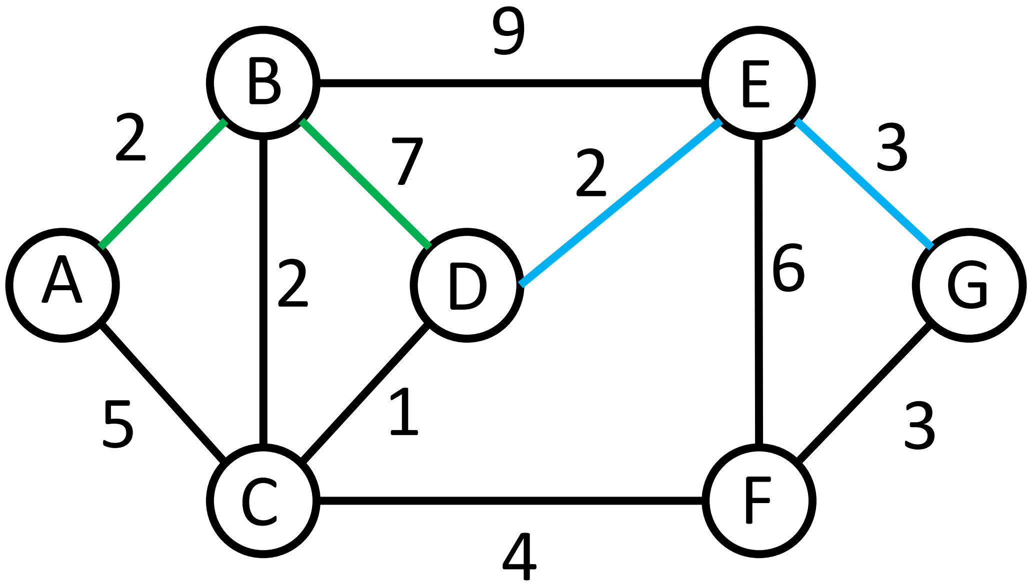


	Cost	Relax.	Best Sol.
	0	10	∞
$X_{A,B} = 1$	2	10	
$X_{B,D} = 1$	9		

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



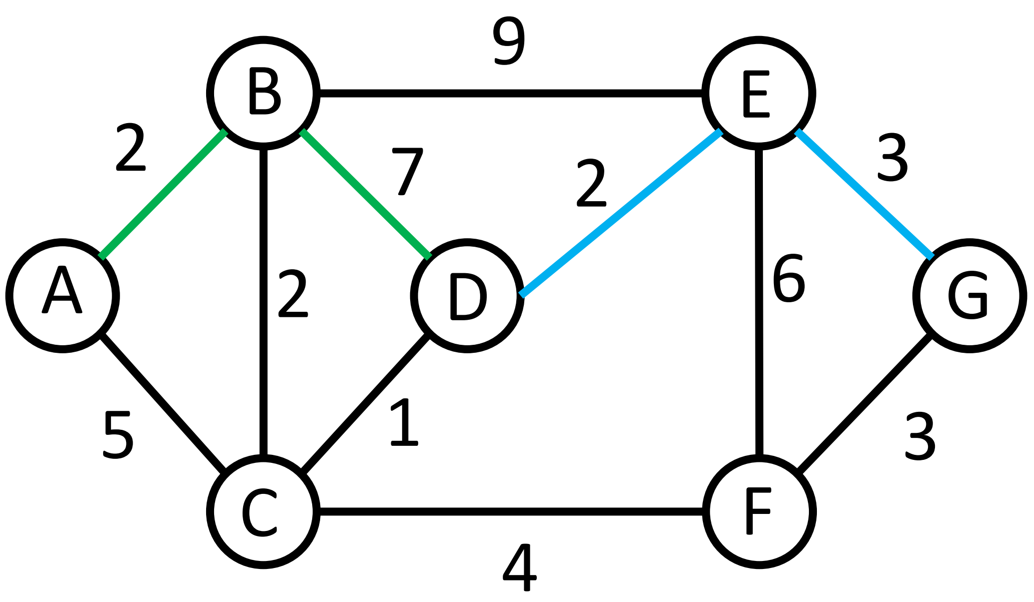
	Cost	Relax.	Best Sol.
	0	10	∞
$X_{A,B} = 1$	2	10	
$X_{B,D} = 1$	9	14	

Relaxed solution feasible!

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



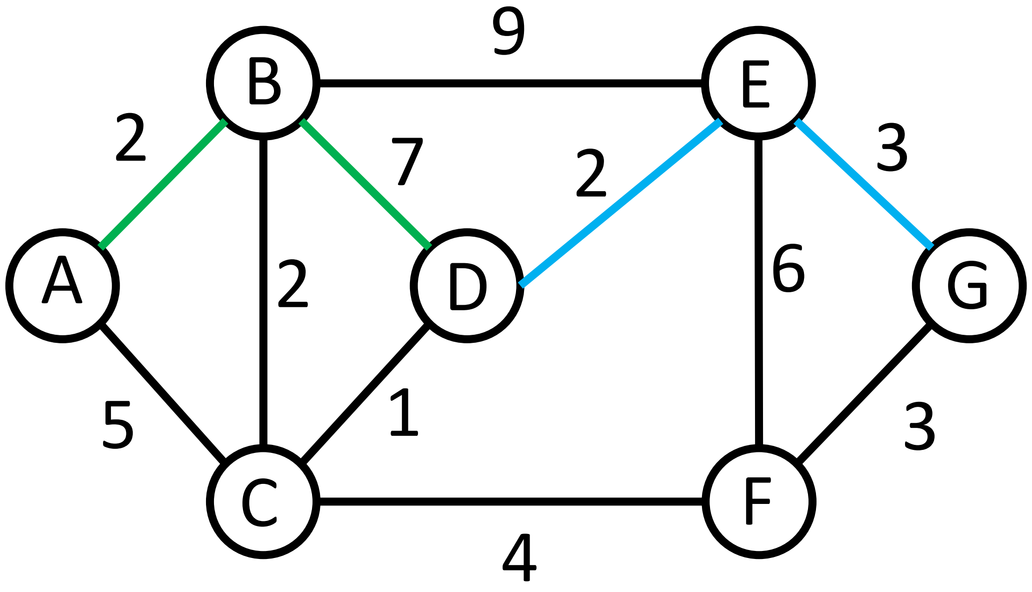
	Cost	Relax.	Best Sol.
	0	10	∞
$X_{A,B} = 1$	2	10	
$X_{B,D} = 1$	9	14	14

Relaxed solution feasible!

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



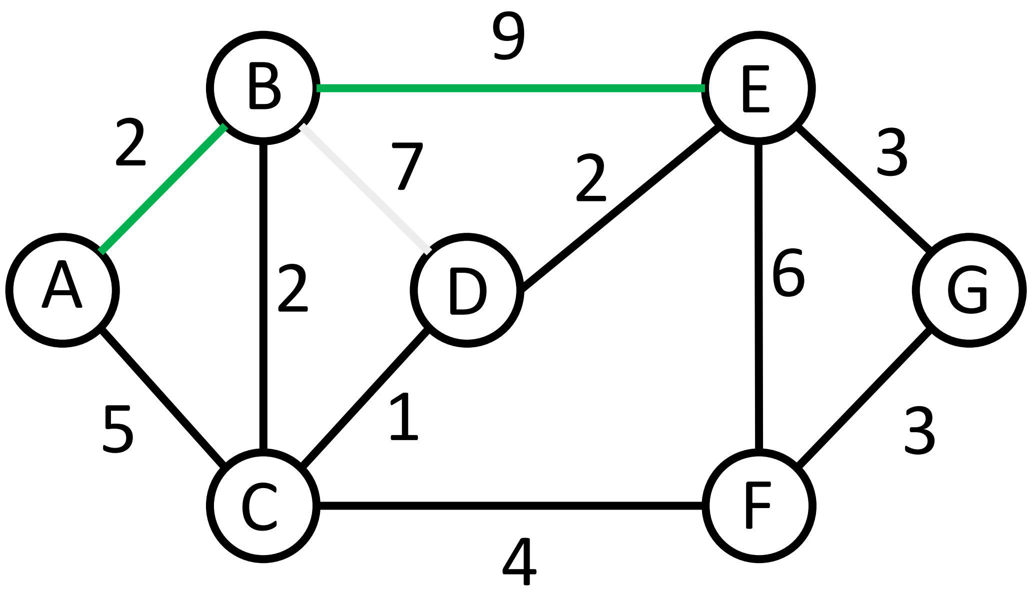
	Cost	Relax.	Best Sol.
	0	10	14
$X_{A,B} = 1$	2	10	
$\cancel{X_{B,D} = 1}$	9	14	14

Backtrack!
The relaxation was feasible, cannot find better solution in this branch

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges

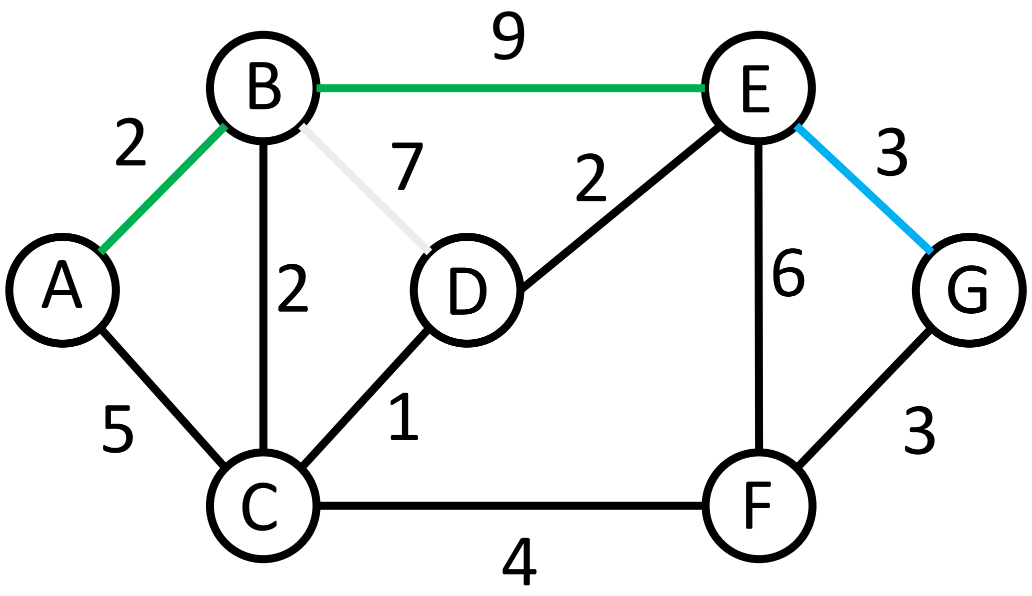


	Cost	Relax.	Best Sol.
	0	10	14
$X_{A,B} = 1$	2	10	
$\cancel{X_{B,D} = 1}$	9	14	14
$X_{B,D} = 0$			
$X_{B,E} = 1$	11		

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



	Cost	Relax.	Best Sol.
	0	10	14
$X_{A,B} = 1$	2	10	
$\cancel{X_{B,D} = 1}$	9	14	14
$X_{B,D} = 0$			
$X_{B,E} = 1$	11	14	

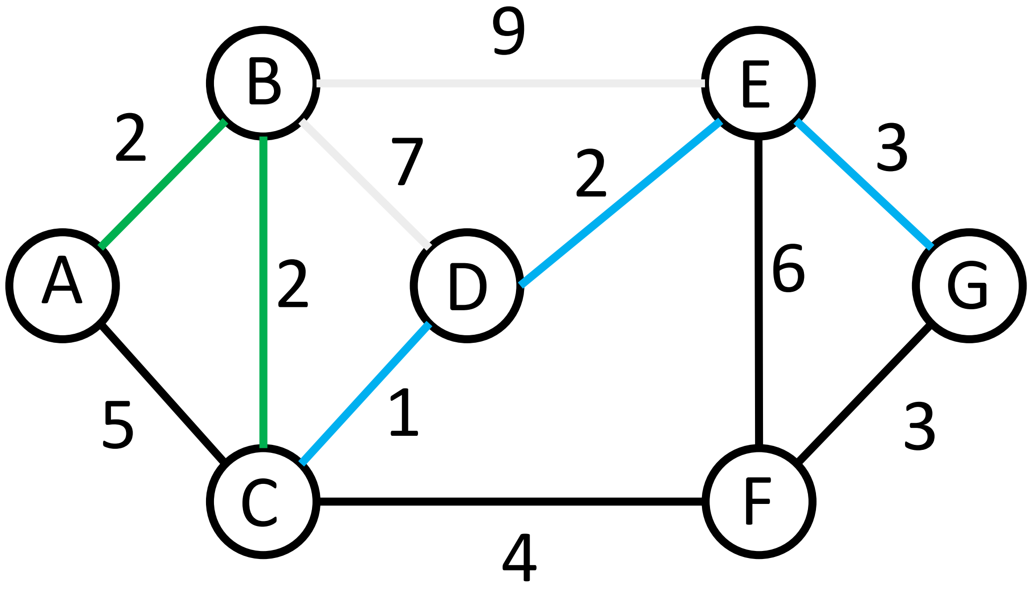
Backtrack!

Relaxation not better than best solution found so far!

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



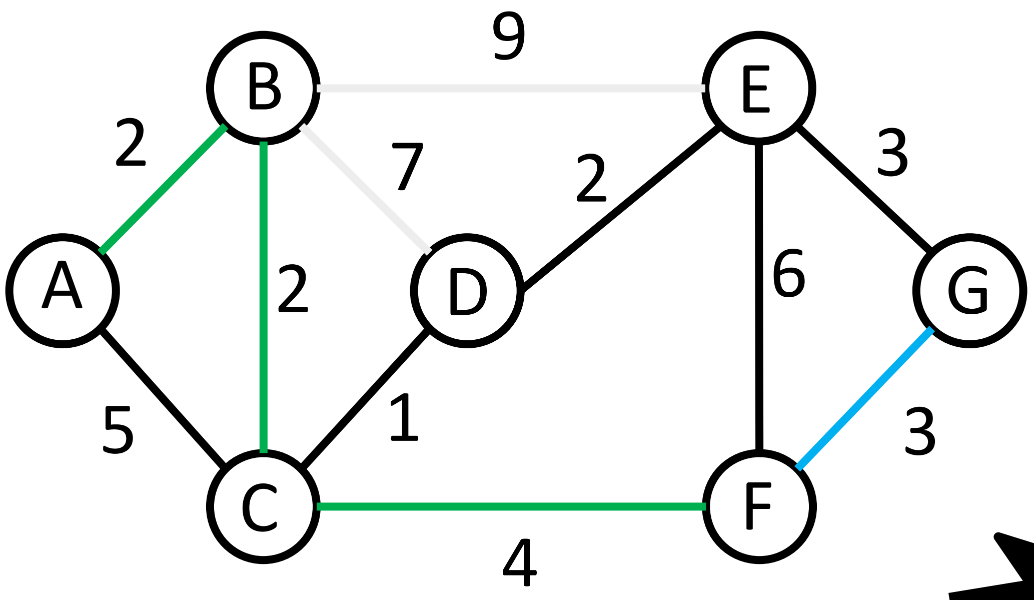
	Cost	Relax.	Best Sol.
	0	10	14
$X_{A,B} = 1$	2	10	
$\cancel{X_{B,D} = 1}$	9	14	14
$X_{B,D} = 0$			
$\cancel{X_{B,E} = 1}$	11	14	
$X_{B,E} = 0$			
$X_{B,C} = 1$	4	10	

Keep going,
Relaxation does not prune

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



	Cost	Relax.	Best Sol.
Use exactly four edges	0	10	14
$X_{A,B} = 1$	2	10	
$\cancel{X_{B,D} = 1}$	9	14	14
$X_{B,D} = 0$			
$\cancel{X_{B,E} = 1}$			
$X_{B,E} = 0$			
$X_{B,C} = 1$	4	10	
$X_{C,F} = 1$	8	11	11

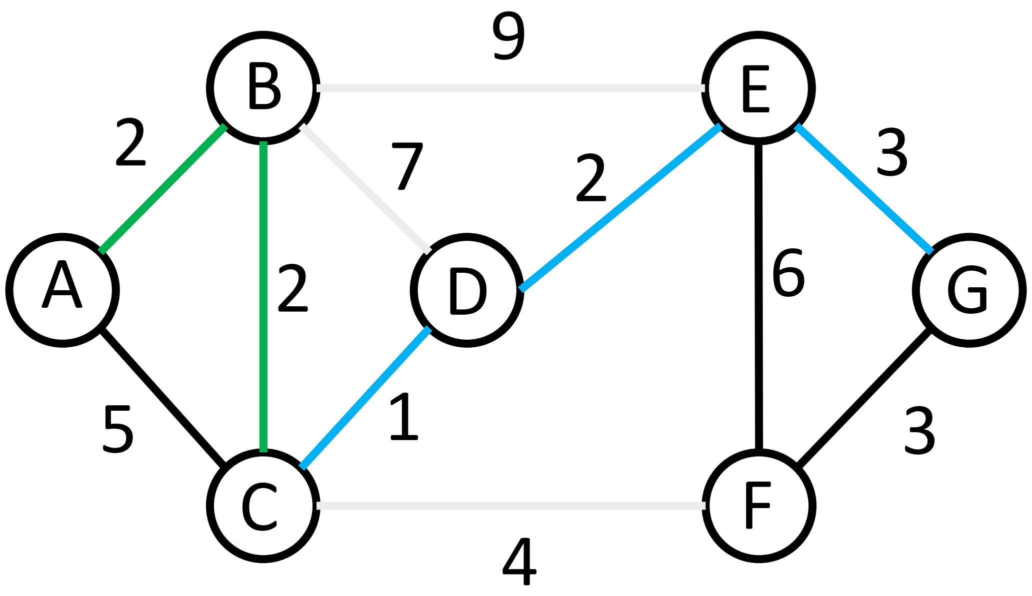
Relaxed solution feasible!

Backtrack

Constrained Shortest Path Problem

Compute shortest path between node A and node G

	Cost	Relax.	Best Sol.
Use exactly four edges	0	10	14
$X_{A,B} = 1$	2	10	
$\cancel{X_{B,D} = 1}$	9	14	14
$X_{B,D} = 0$			
$\cancel{X_{B,E} = 1}$	11	14	
$X_{B,E} = 0$			
$X_{B,C} = 1$	4	10	
$\cancel{X_{C,F} = 1}$	8	11	11
$X_{C,F} = 0$			

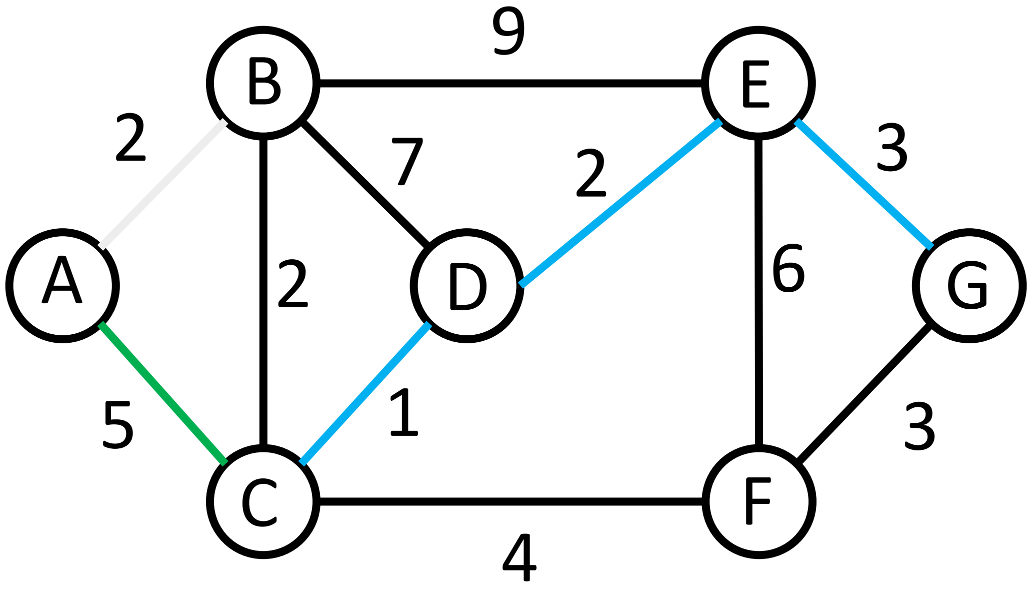


However the relaxation is 11,
not better than the best solution, backtrack!

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



	Cost	Relax.	Best Sol.
	0	10	11
$\cancel{X_{A,B} = 1}$	2	10	11
$X_{A,B} = 0$			
$X_{A,C} = 1$	5	11	

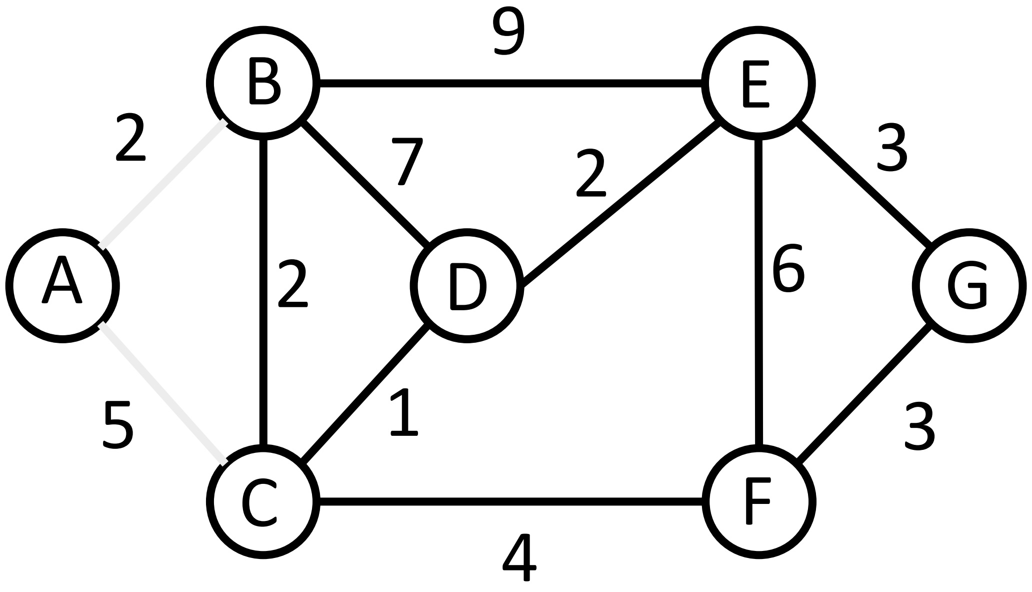
Backtrack!

Relaxation is equal or greater than the best solution

Constrained Shortest Path Problem

Compute shortest path between node A and node G

Use exactly four edges



	Cost	Relax.	Best Sol.
	0	10	11
$\cancel{X_{A,B} = 1}$	2	10	11
$X_{A,B} = 0$			
$\cancel{X_{A,C} = 1}$	5	12	
$X_{A,C} = 0$			

Done, no options left!

Best solution found has cost 11

(Exhaustive) Search

Iteratively extend a feasible partial solution

Sophisticated brute force

Guaranteed to
find (optimal) solution

Slow
when inference is weak

(Exhaustive) Search

Iteratively extend a feasible partial solution

Sophisticated brute force

Guaranteed to
find (optimal) solution

Slow
when inference is weak

Local Search

Iteratively changing an infeasible solution

Much faster
when inference is weak

No guarantees
to find (optimal) solution

Not in this block!
(mainly FYI)

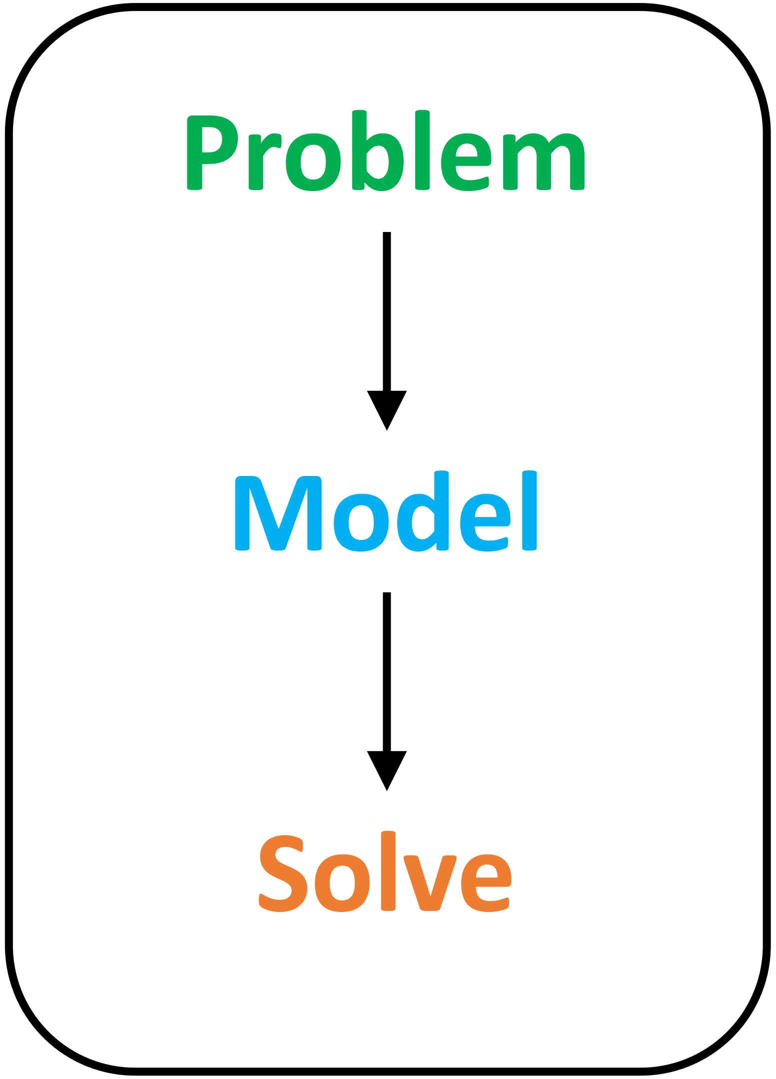
Local Search

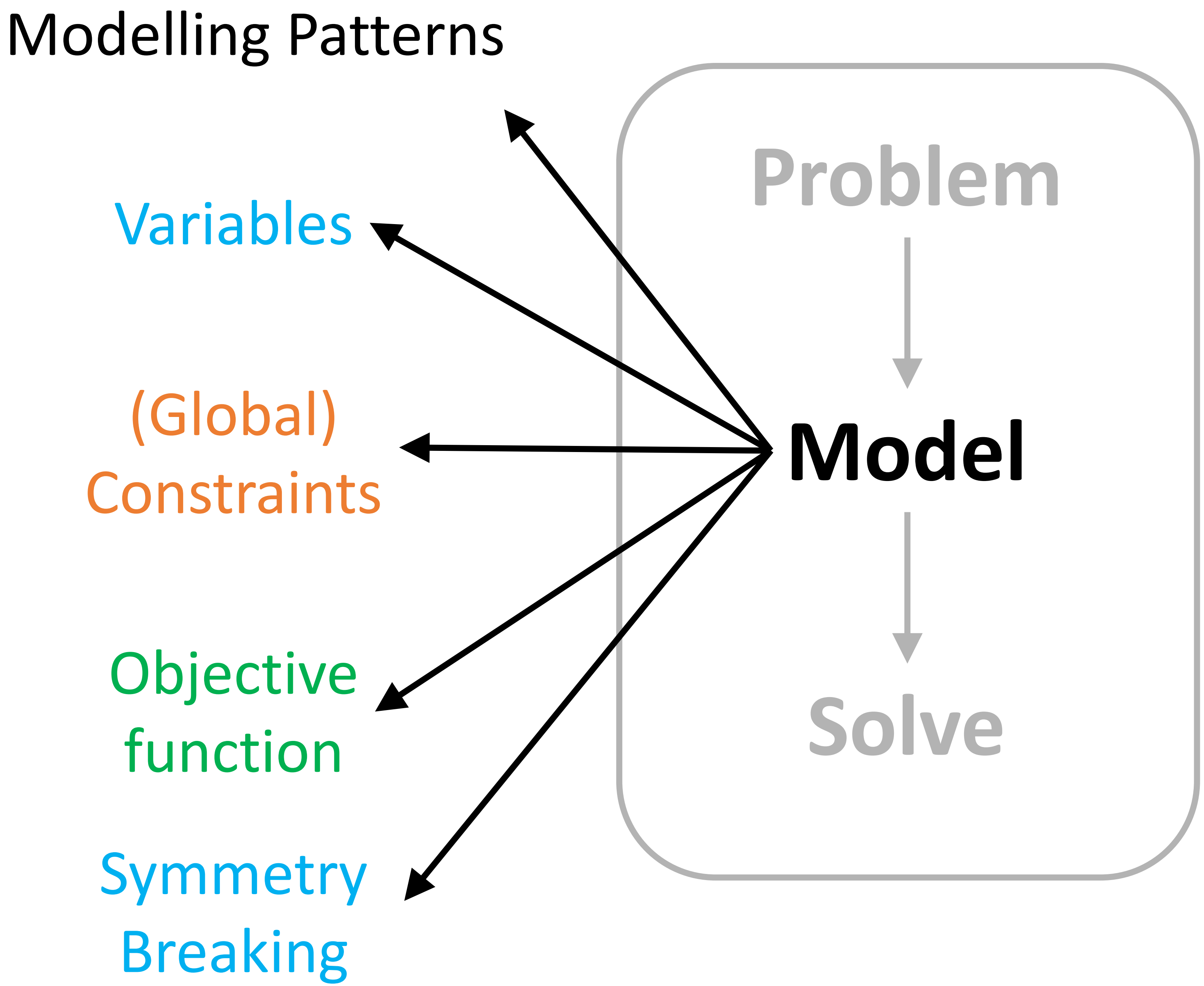
Iteratively changing an infeasible solution

Much faster
when inference is weak

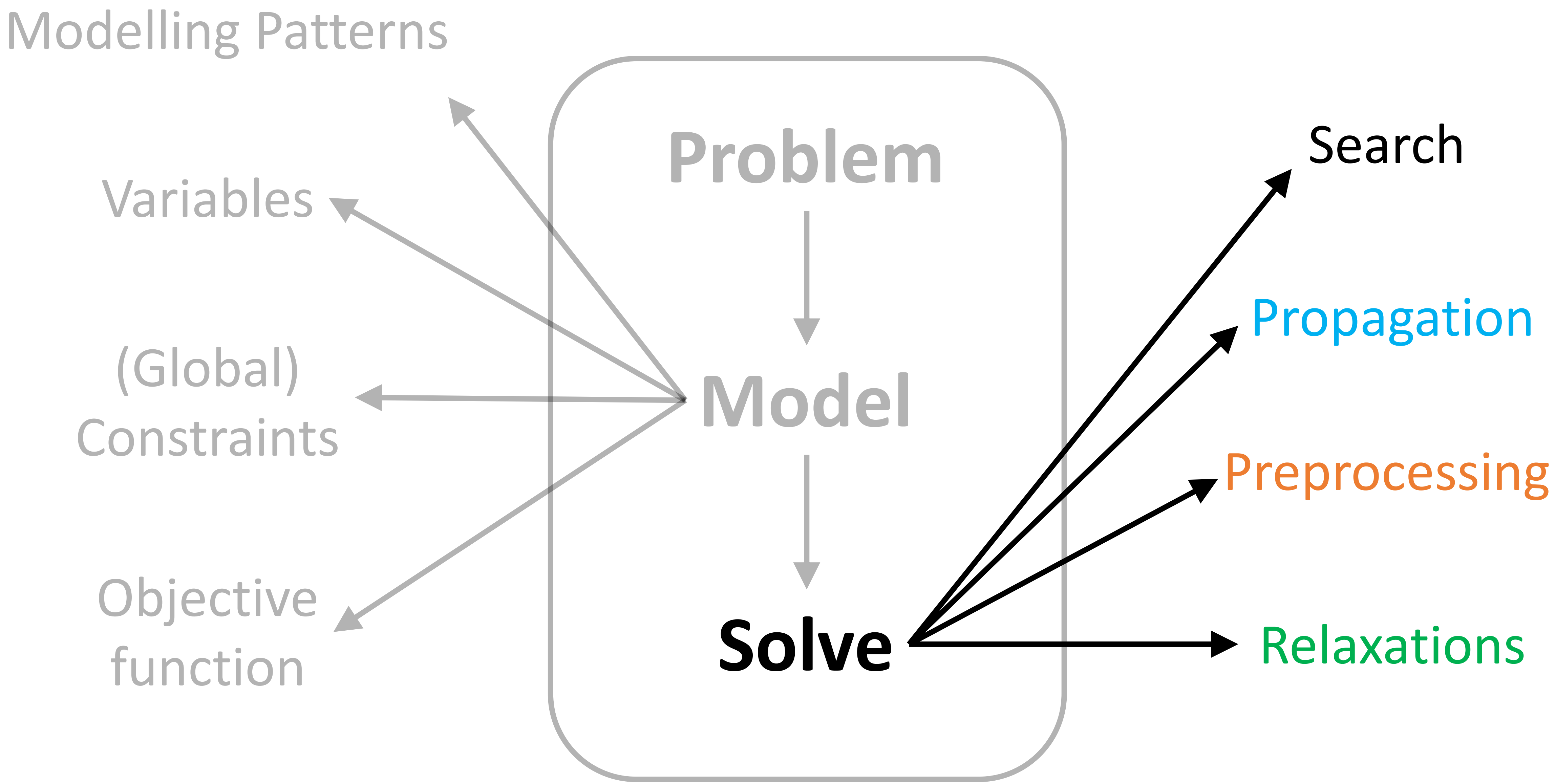
No guarantees
to find (optimal) solution

Summary of Combinatorial Optimisation Block



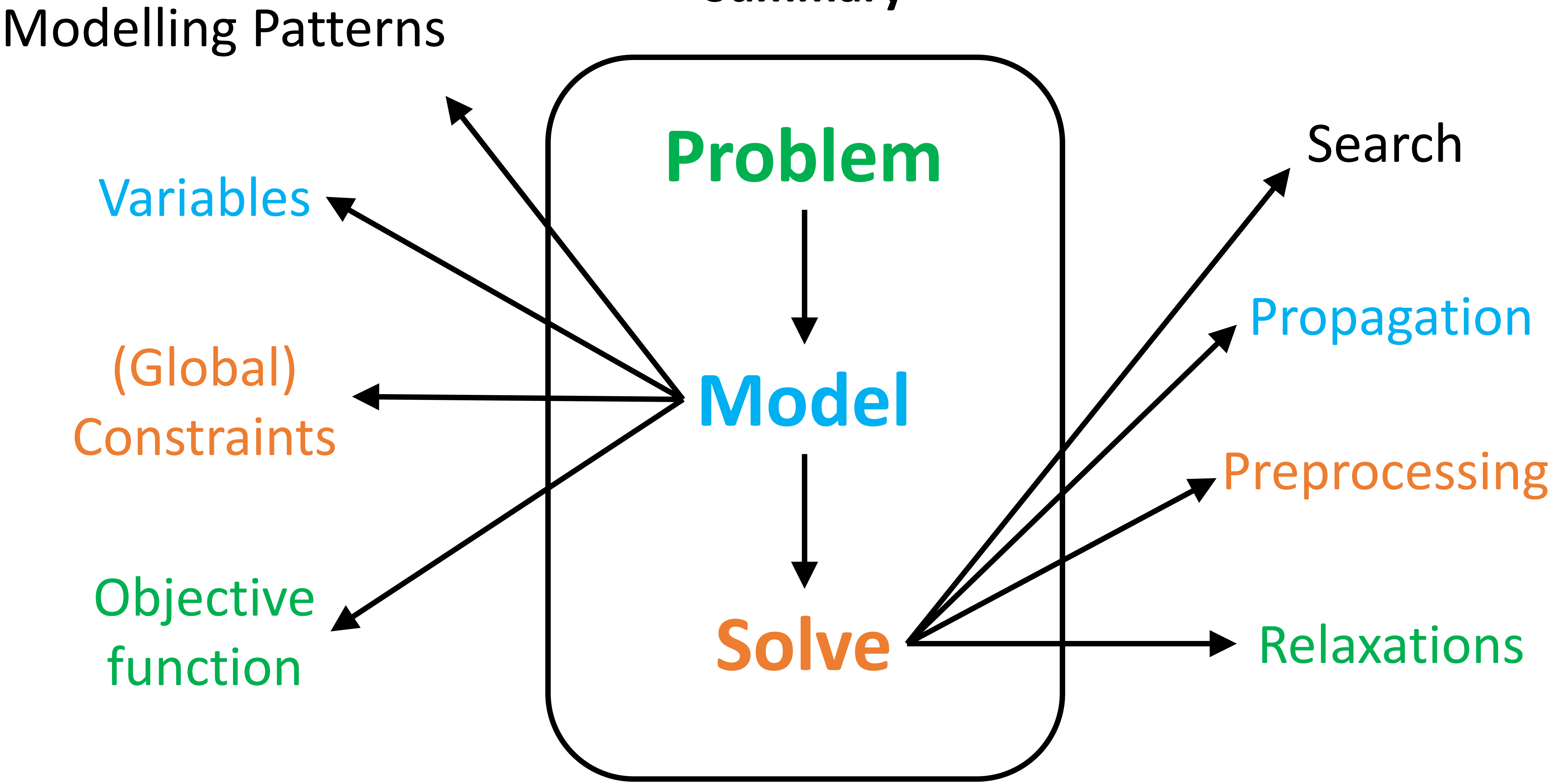


Modelling Patterns
Variables
(Global)
Constraints
Objective
function
Symmetry
Breaking
Problem
Model
Solve



Modelling Patterns
Variables
(Global) Constraints
Objective function
Problem
Model
Solve
Search
Propagation
Preprocessing
Relaxations

Summary


Modelling Patterns
Variables
(Global) Constraints
Objective function
Problem
Model
Solve
Search
Propagation
Preprocessing
Relaxations

Post-Lecture Update

Q: Why is the constrained shortest path NP-Hard?

A: We can prove that the constrained shortest path problem (CSPP) is NP-hard by showing that we can solve Travelling Salesperson problme (TSP), a classic NP-hard problem, by converting it (in polynomial time) into CSPP. This is called a polynomial reduction and is a popular way of showing hardness.

The TSP takes as input a graph G = (V, E) with weighted edges and asks for a cycle of minimum cost that visits each city exactly once.

We can use CSPP to solve TSP as follows. For the CSPP, we construct a directed graph G' based on the TSP graph. For each vertex $v_i \in V$, we create two vertices v_i^{in} and v_i^{out}, and add a dummy source and sink vertices. For each edge $(v_i, v_j) \in E$, we add edges to the new graph (v_i^{out}, v_j^{in}). We also add edges (v_i^{in}, v_i^{out}) with weight 0, and edges (source, v_i^{in}) and (v_i^{out}, sink) with weights 0. To solve the TSP, we solve the newly constructed CSPP by asking for the minimum cost path between the source and the sink with exactly 2n+2 edges, where 'n' is the number of nodes in original graph (|V|). Note that we cannot visit the same node more than once, since from vertex v_i^{in} there is only one edges towards v_i^{out}, and in CSPP we cannot use the same edge more than once (we seek a simple path). Indeed the minimum cost CSPP solution can be translated back to the TSP with the same cost.

The above can also be seen as a modelling exercise! Note that you may still be able to solve the CSPP with dynamic programming, even if it is NP-hard. But your DP will likely not be polynomial. For instance, the DP may have runtime O(n^k), where k is the exact number of edges. Since k can be set to $k = \frac{n}{2}$, we see the runtime is exponential. Another example of a DP for an NP-hard problem is the Knapsack problem.

Say you claim that you do have a polynomial algorithm to solve the constrained shortest path. Assuming my reduction is correct, then either 1) you solved the greatest open question of modern computer science "Is P=NP?", or 2) your algorithm is not actually polynomial!

Post-Lecture Update

Q: How do we implement global constraints, e.g., the shortest path propagator? Do we recompute each iteration, sounds expensive!

In practice, it is important to implement propagators well. Incrementality is a key feature of any propagator: given a small change to the domain, the propagator should also do only a small amount of work. Otherwise, it is impractical, e.g., indeed recomputing the shortest path every iteration will likely dominant the runtime.

Note that for this specific propagator, we could opt to precompute distances between nodes. However, this is a much-simplified version of the propagator. During the search, it is expected that due to backtracking and pruning, many edges will effectively get removed from the graph. In these cases, the precomputed distances may no longer be accurate values for the distance, although they could still serve as a lower bound for the distances.

For this course it is enough to acknowledge that incrementality is important, but we do not go into the details. We cover these details a bit more in the master course Algorithms for Intelligent Decision Making.

On a related note, if you would like to look up the shortest path propagator, you may be surprised how it is done! You may read up on the details in the PhD thesis of Diego de Uña https://minerva-access.unimelb.edu.au/items/f7d71a4b-1dc4-583e-aebe-d5b5a4965954