

Constraint Satisfaction Problems (CSPs)

CS 221 Section

Richard Diehl Martinez

Agenda

- CSP Problem Modeling
- N-ary Constraints
- Elimination Example

- CSP Problem Modeling
- N-ary Constraints
- Elimination Example

Definition: Factor Graph

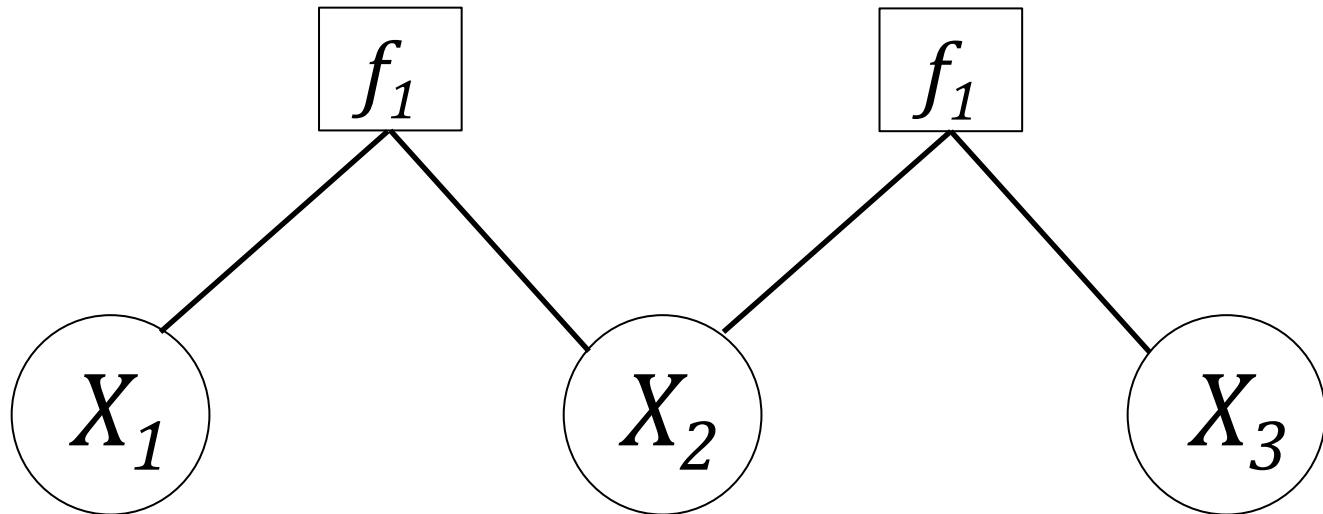
Variables:

$X = (X_1, \dots, X_n)$, $X_i \in \text{Domain}_i$ where

Factors:

f_1, \dots, f_m ,

with each $f_j(X) \geq 0$



Definition: Constraint Satisfaction Problem (CSP)

A CSP is a factor graph where all factors are **constraints**:

for all $j = 1, \dots, m$.

The constraint is satisfied iff $f_j(x) = 1$.

Definition: Consistent Assignments

An assignment x if $\text{Weight}(x) = 1$ (i.e., all constraints are satisfied.)

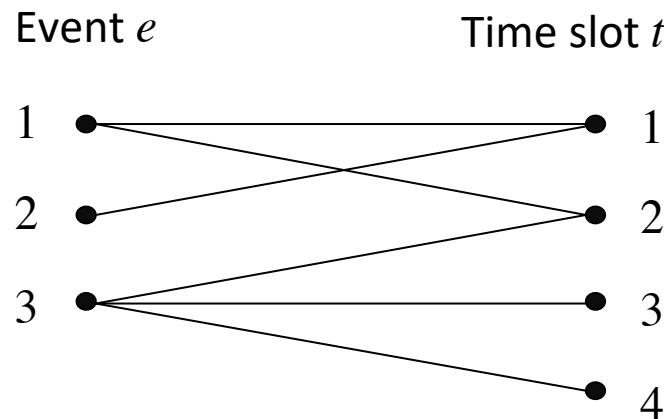
Factor Graph and CSP Applications

- Scheduling problems: event scheduling, resource and assembly scheduling
- Inferring relations from data
- Puzzles: sudoku, crosswords
- Satisfiability problems
- Map and graph coloring
- Object tracking
- Decoding noisy signals (images, messages etc.)

Event Scheduling

Setup:

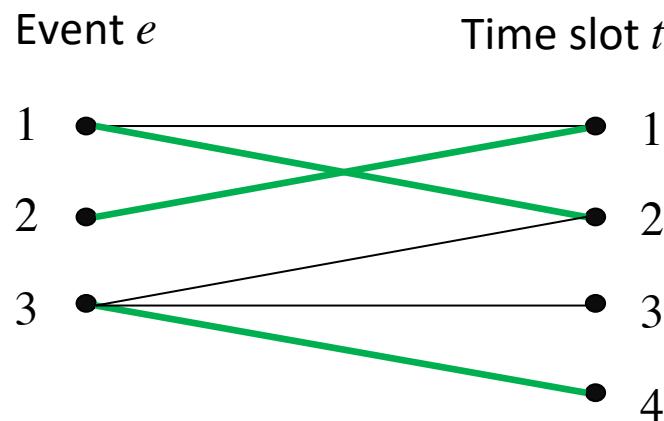
- Have E events and T time slots
- Each event e must be put in **exactly one** time slot
- Each time slot t can have **at most one** event
- Event e only allowed at time slot t if (e, t) in A



Event Scheduling

Setup:

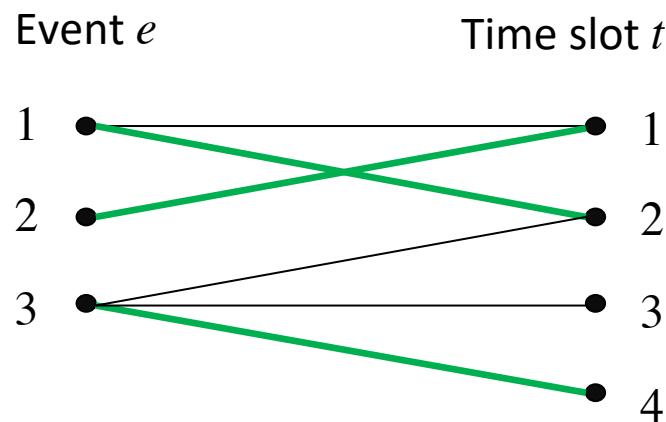
- Have E events and T time slots
- Each event e must be put in **exactly one** time slot
- Each time slot t can have **at most one** event
- Event e only allowed at time slot t if (e, t) in A



Event Scheduling

Formulation 1a:

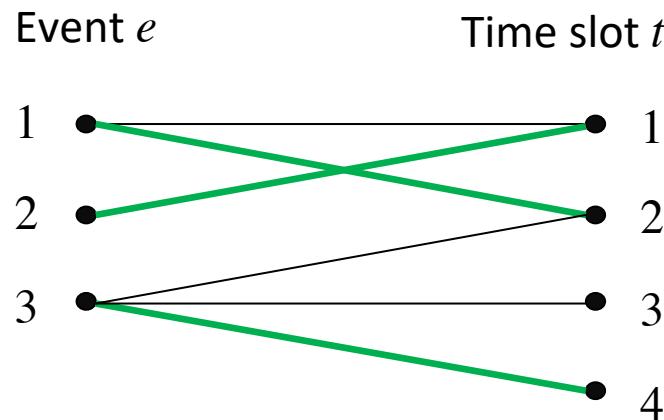
- Variables for each event e , $X_e \in \{1, \dots, T\}$



Event Scheduling

Formulation 1a:

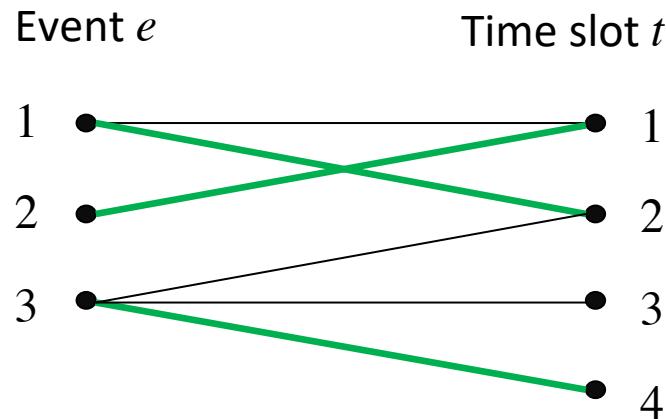
- Variables for each event e , $X_e \in \{1, \dots, T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$



Event Scheduling

Formulation 1a:

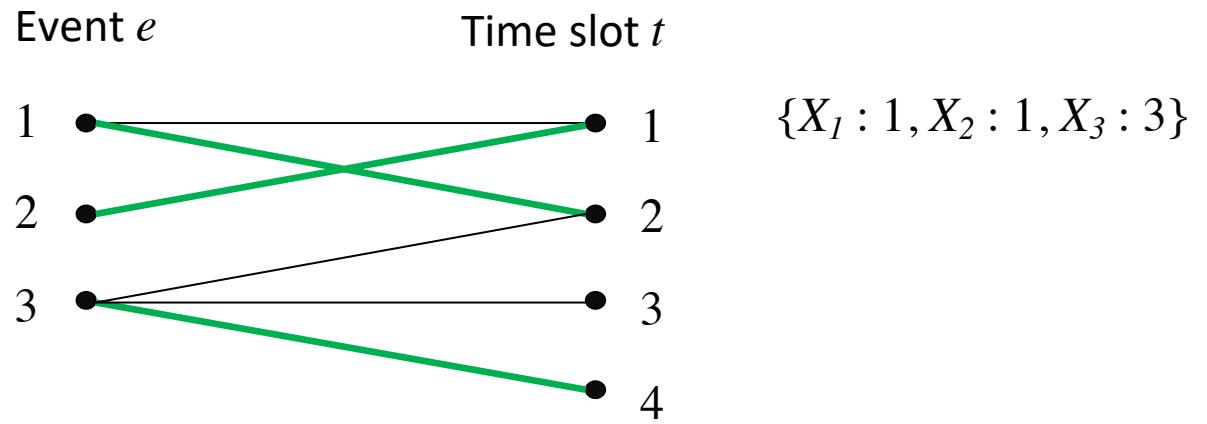
- Variables for each event e , $X_e \in \{1, \dots, T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$
- Constraints (only schedule allowed times): for each event e , enforce $[(e, X_e) \in A]$



Event Scheduling

Formulation 1a:

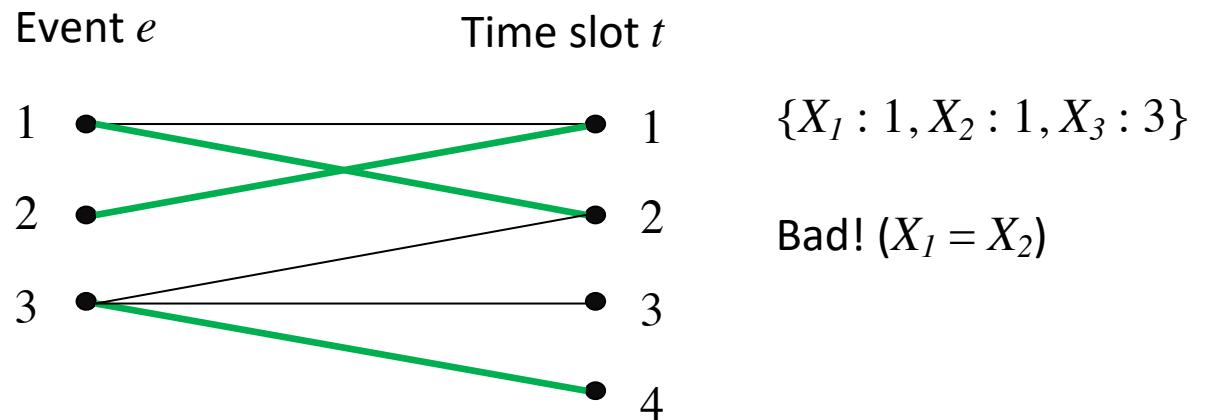
- Variables for each event e , $X_e \in \{1, \dots, T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$
- Constraints (only schedule allowed times): for each event e , enforce $[(e, X_e) \in A]$



Event Scheduling

Formulation 1a:

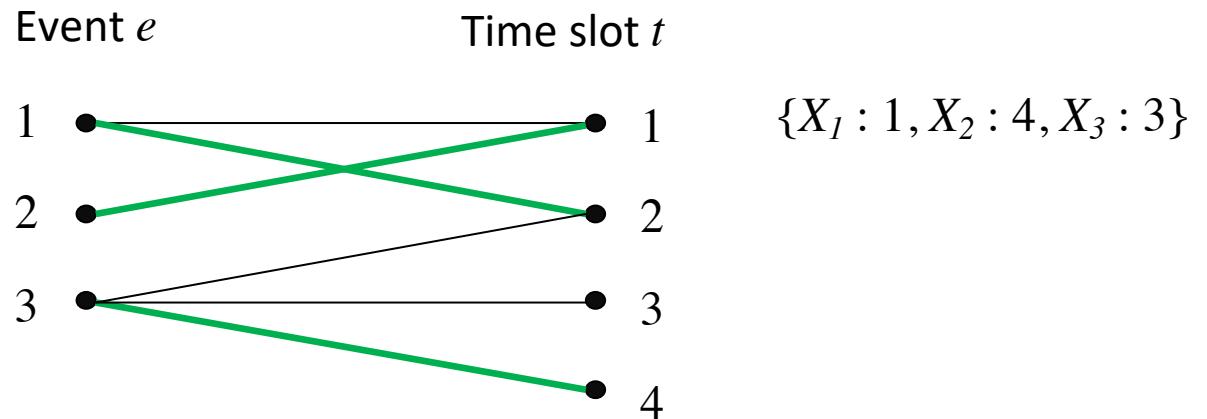
- Variables for each event e , $X_e \in \{1, \dots, T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$
- Constraints (only schedule allowed times): for each event e , enforce $[(e, X_e) \in A]$



Event Scheduling

Formulation 1a:

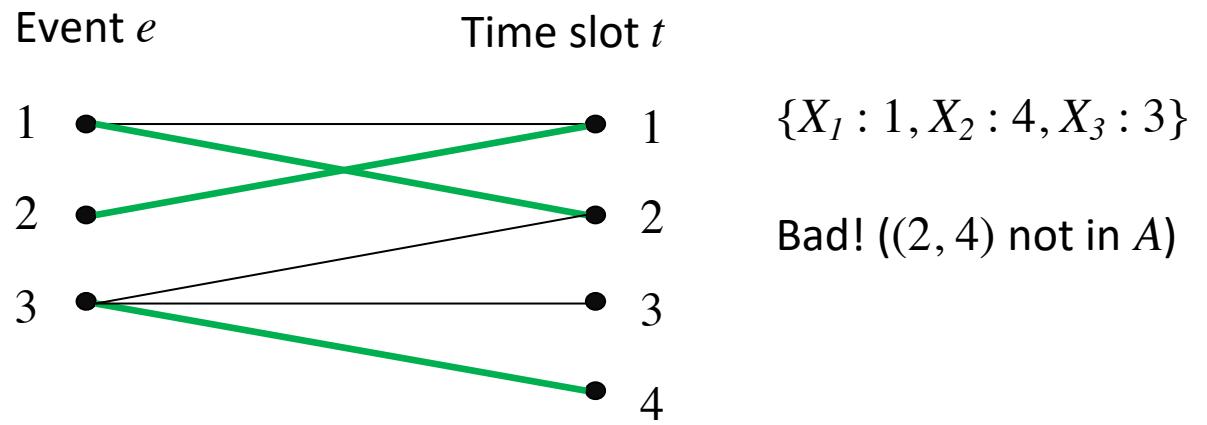
- Variables for each event e , $X_e \in \{1, \dots, T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$
- Constraints (only schedule allowed times): for each event e , enforce $[(e, X_e) \in A]$



Event Scheduling

Formulation 1a:

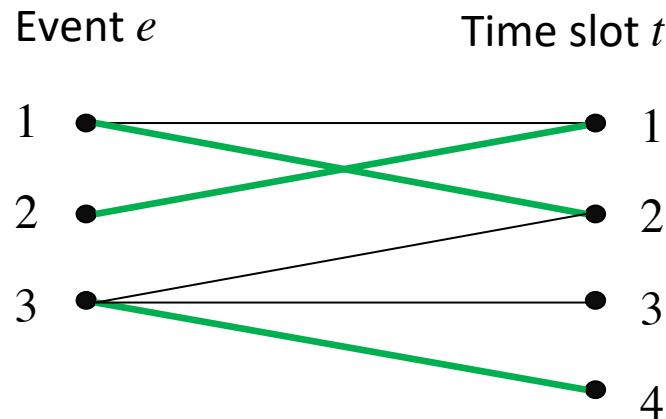
- Variables for each event e , $X_e \in \{1, \dots, T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$
- Constraints (only schedule allowed times): for each event e , enforce $[(e, X_e) \in A]$



Event Scheduling

Formulation 1a:

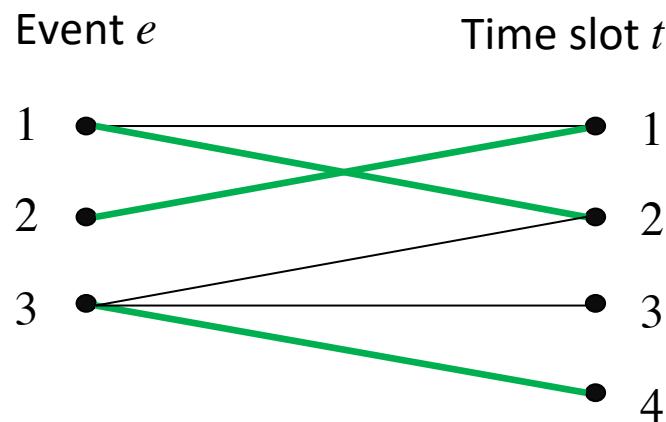
- Variables for each event e , $X_e \in \{1, \dots, T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$
- Constraints (only schedule allowed times): for each event e , enforce $[(e, X_e) \in A]$



Event Scheduling

Formulation 1b:

- Variables for each event e , X_1, \dots, X_E

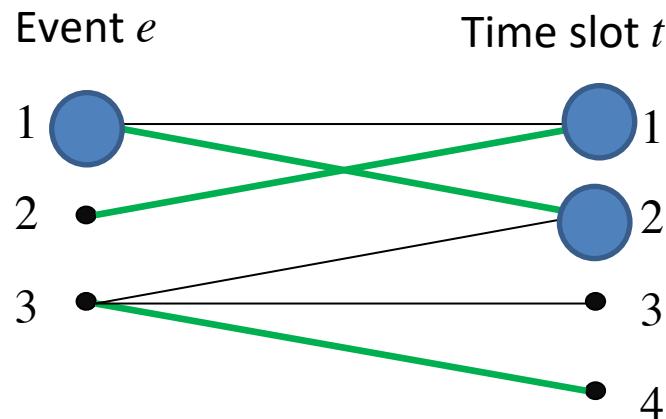


Event Scheduling

Formulation 1b:

- Variables for each event e , X_1, \dots, X_E

$$Domain_i = \{t : (i, t) \in A\}$$



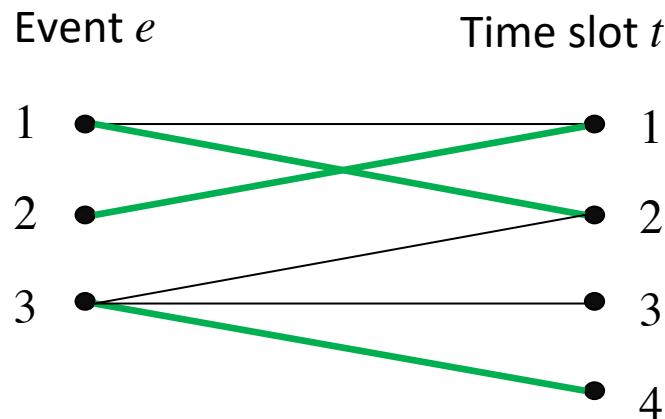
Event Scheduling

Formulation 1b:

- Variables for each event e , X_1, \dots, X_E

$$Domain_i = \{t : (i, t) \in A\}$$

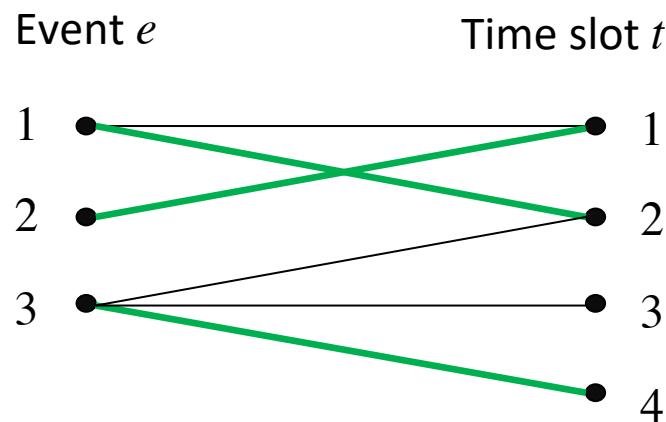
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$



Event Scheduling

Formulation 2a:

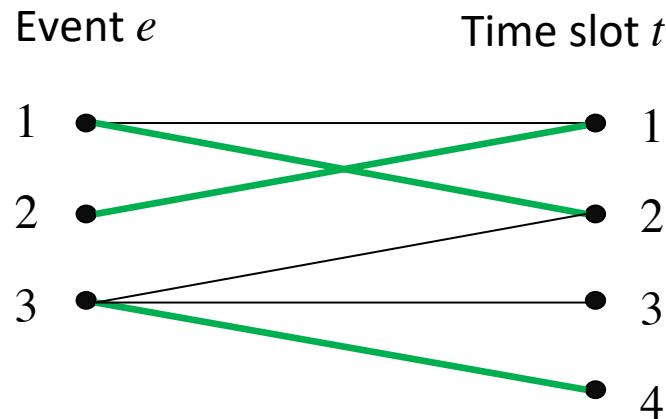
- Variables for each time slot t : $Y_t \in \{1, \dots, E\} \cup \{\emptyset\}$



Event Scheduling

Formulation 2a:

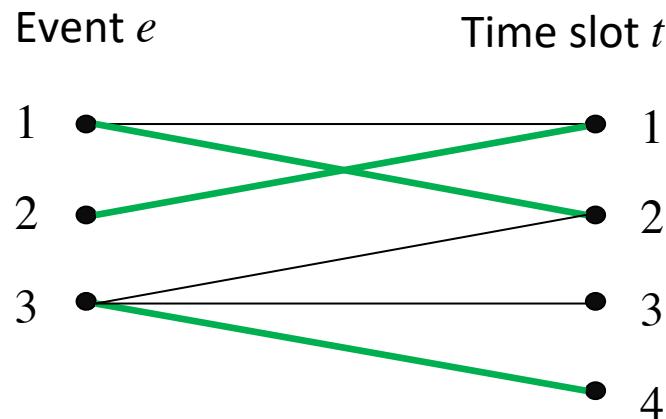
- Variables for each time slot t : $Y_t \in \{1, \dots, E\} \cup \{\emptyset\}$
- Constraints (each event is scheduled exactly once): for each event e , enforce [$Y_t = e$ for exactly one t]



Event Scheduling

Formulation 2a:

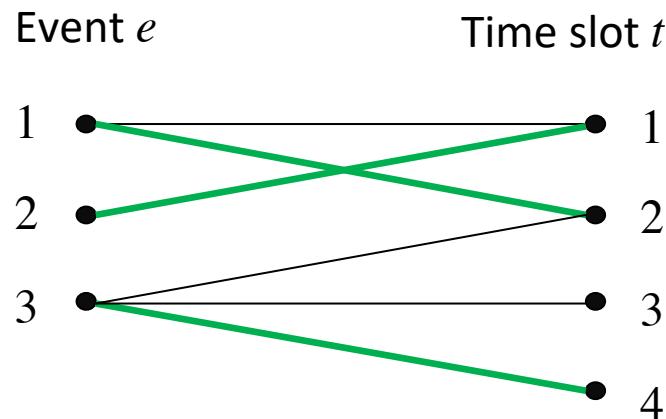
- Variables for each time slot t : $Y_t \in \{1, \dots, E\} \cup \{\emptyset\}$
- Constraints (each event is scheduled exactly once): for each event e , enforce $[Y_t = e \text{ for exactly one } t]$
- Constraints (only schedule allowed times): for each time slot t , enforce $[Y_t = \emptyset \text{ or } (Y_t, t) \in A]$



Event Scheduling

Formulation 2a:

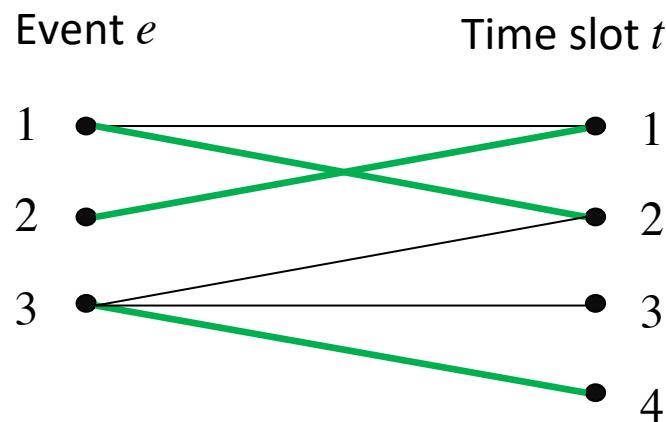
- Variables for each time slot t : $Y_t \in \{1, \dots, E\} \cup \{\emptyset\}$
- Constraints (each event is scheduled exactly once): for each event e , enforce $[Y_t = e \text{ for exactly one } t]$
- Constraints (only schedule allowed times): for each time slot t , enforce $[Y_t = \emptyset \text{ or } (Y_t, t) \in A]$



Event Scheduling

Formulation 2a:

- Variables for each time slot t : Y_1, \dots, Y_T

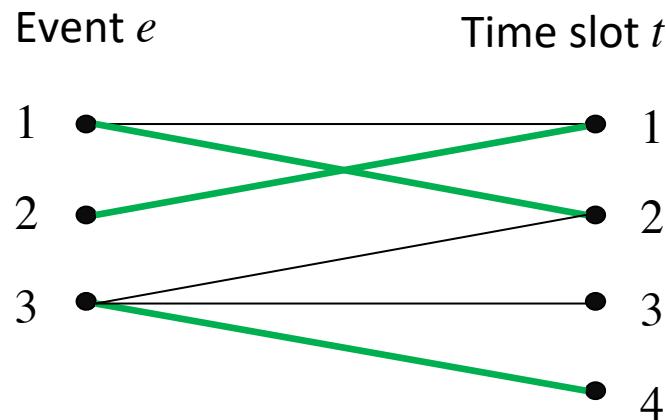


Event Scheduling

Formulation 2a:

- Variables for each time slot t : Y_1, \dots, Y_T

$$Domain_i = \{e : (e, i) \in A\} \cup \{\emptyset\}$$



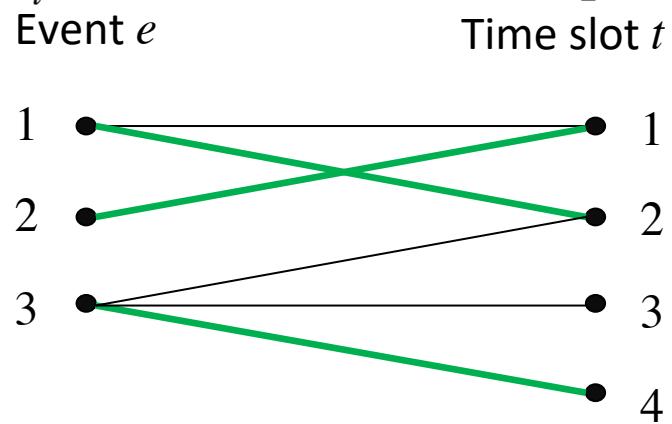
Event Scheduling

Formulation 2a:

- Variables for each time slot t : Y_1, \dots, Y_T

$$Domain_i = \{e : (e, i) \in A\} \cup \{\emptyset\}$$

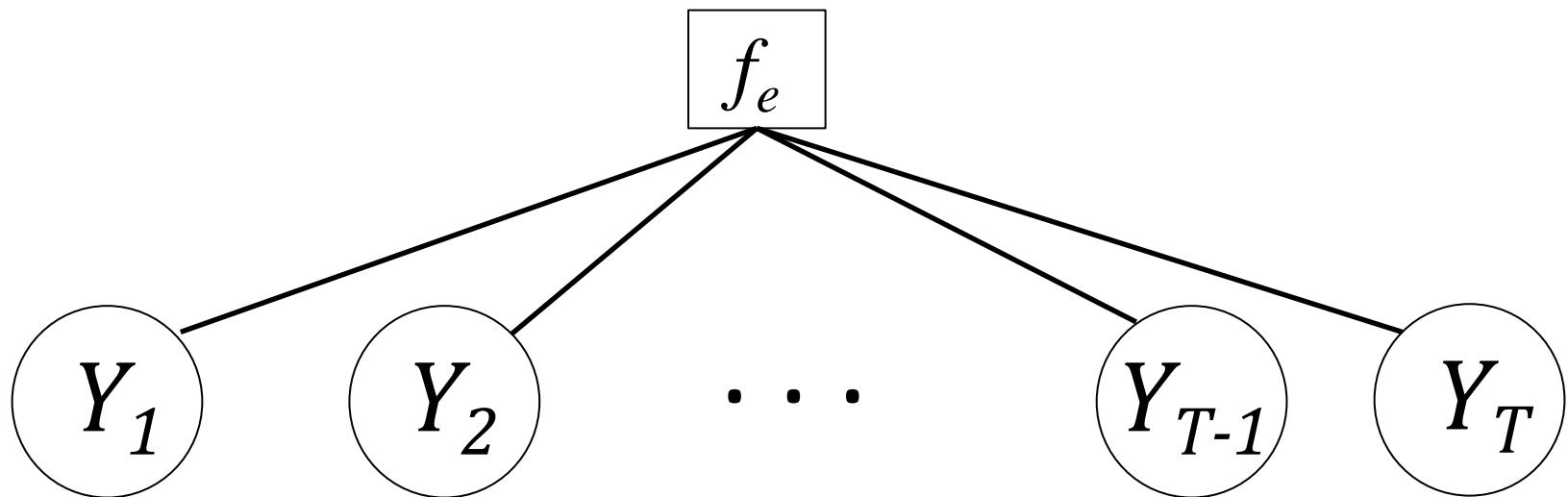
- Constraints (each event is scheduled exactly once): for each event e , enforce [$Y_t = e$ for exactly one t]



- Problem Modeling
- N-ary Constraints
- Elimination Example

N-ary Constraints

- From event scheduling:
 - Constraints (each event is scheduled exactly once): for each event e , enforce
 $[Y_t = e \text{ for exactly one } t]$



N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0				

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

Processing: $[A_i = A_{i-1} + 1[Y_i = e]]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0				

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

Processing: $[A_i = \min(A_{i-1} + 1[Y_i = e], 2)]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0				

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

Processing: $[A_i = \min(A_{i-1} + 1[Y_i = e], 2)]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0	0			

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

Processing: $[A_i = \min(A_{i-1} + 1[Y_i = e], 2)]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0	0	1		

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

Processing: $[A_i = \min(A_{i-1} + 1[Y_i = e], 2)]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0	0	1	1	

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

Processing: $[A_i = \min(A_{i-1} + 1[Y_i = e], 2)]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0	0	1	1	2

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

Processing: $[A_i = \min(A_{i-1} + 1[Y_i = e], 2)]$

Final Output: $1[A_T = 1]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0	0	1	1	2

N-ary Constraints

Key Idea: Auxiliary Variables

Auxiliary Variables hold intermediate computation.

Represent “for exactly one” as counting the number of values equal to e and constraining that count to be equal to one.

Factors:

Initialization: $[A_0 = 0]$

Processing: $[A_i = \min(A_{i-1} + 1[Y_i = e], 2)]$

Final Output: $1[A_T = 1]$

i	0	1	2	3	4
Y_i		3	1	2	1
A_i	0	0	1	1	2

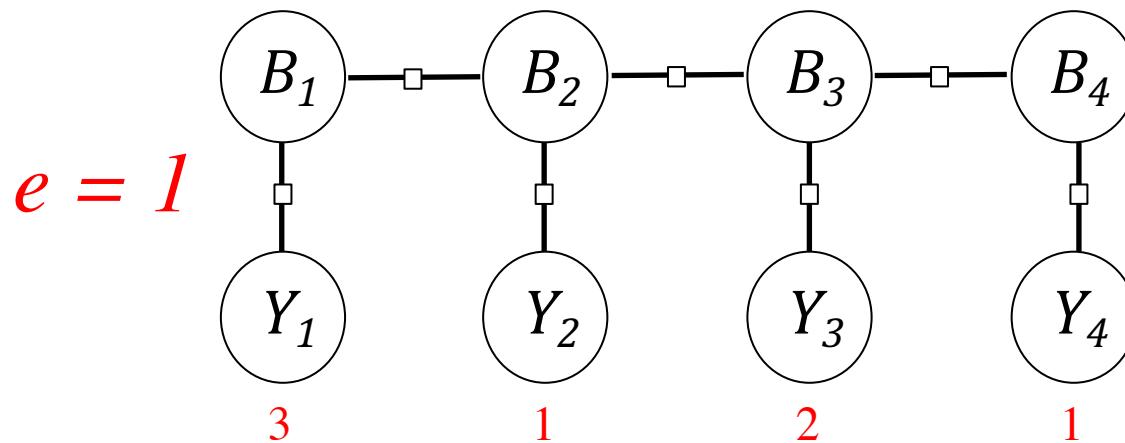
Still have factors with three variables...

N-ary Constraints

Key idea: Combine A_{i-1} and A_i into one variable B_i

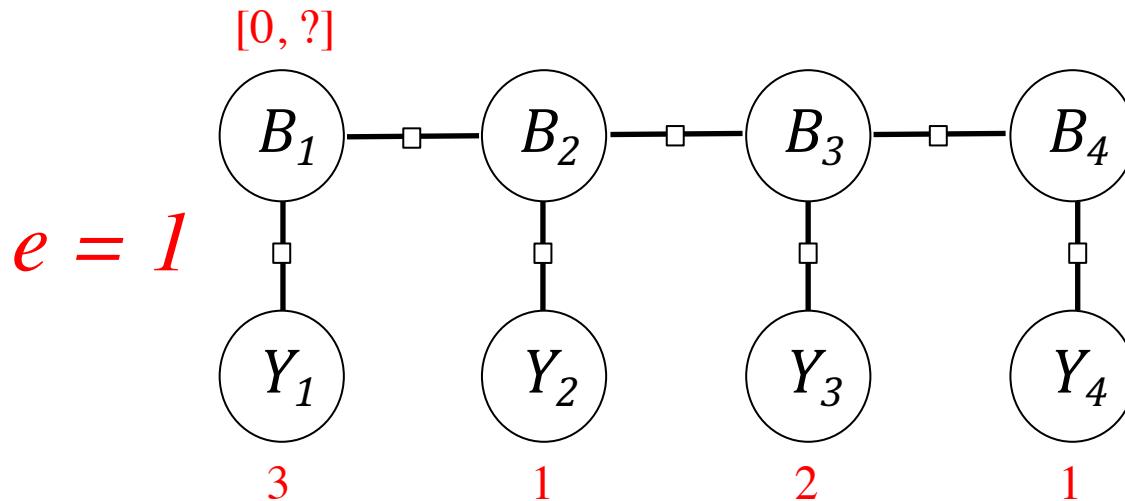
N-ary Constraints

Key idea: Combine A_{i-1} and A_i into one variable B_i



N-ary Constraints

Key idea: Combine A_{i-1} and A_i into one variable B_i

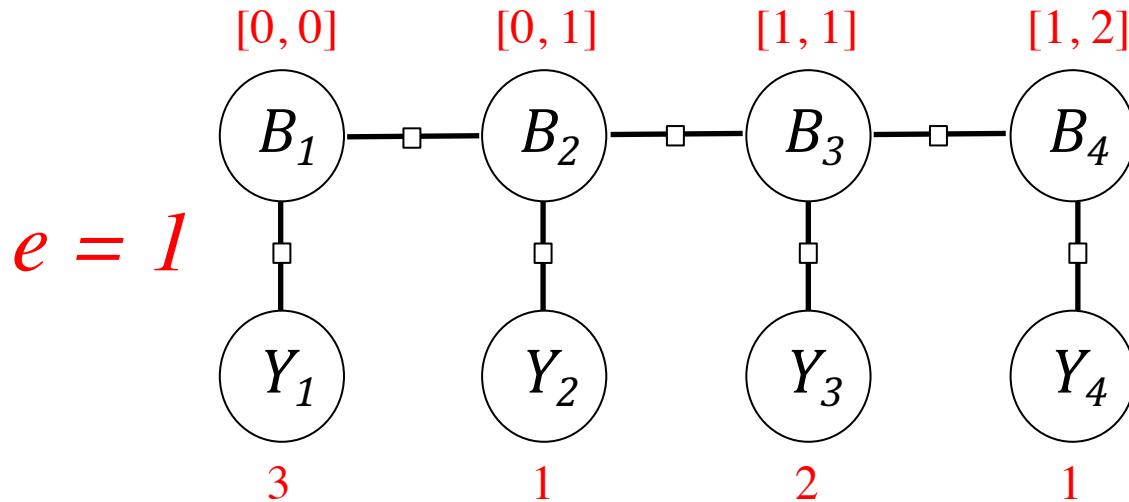


Factors:

Initialization: $[B_1[0] = 0]$

N-ary Constraints

Key idea: Combine A_{i-1} and A_i into one variable B_i



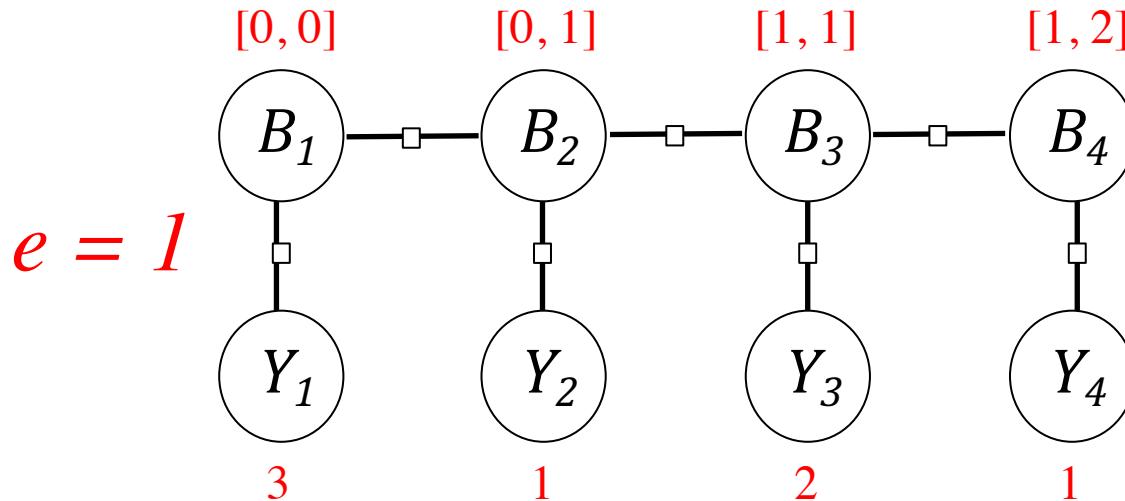
Factors:

Initialization: $[B_1[0] = 0]$

Processing: $[B_i[1] = \min(B_i[0] + 1[Y_i = e], 2)]$

N-ary Constraints

Key idea: Combine A_{i-1} and A_i into one variable B_i



Factors:

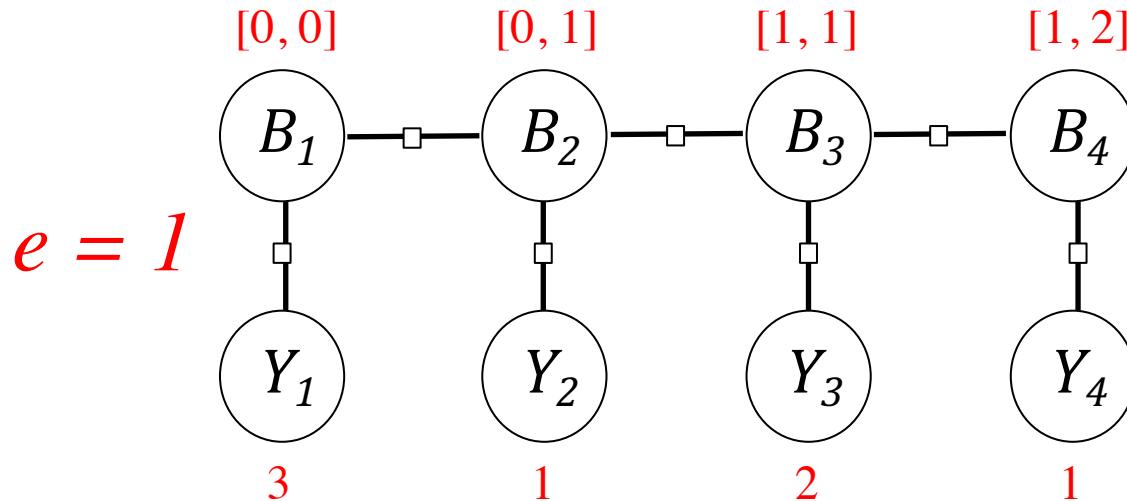
Initialization: $[B_1[0] = 0]$

Processing: $[B_i[1] = \min(B_i[0] + 1[Y_i = e], 2)]$

Final Output: $1[B_T[1] = 1]$

N-ary Constraints

Key idea: Combine A_{i-1} and A_i into one variable B_i



Factors:

Initialization: $[B_1[0] = 0]$

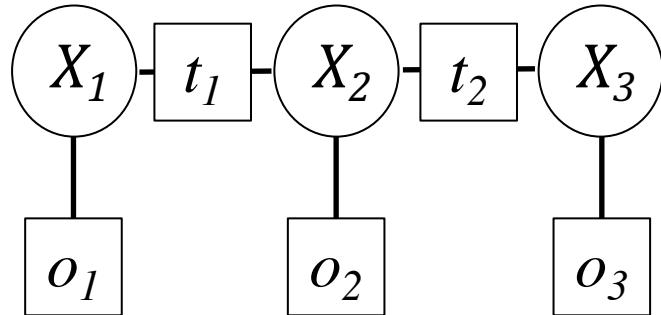
Processing: $[B_i[1] = \min(B_i[0] + 1[Y_i = e], 2)]$

Final Output: $1[B_T[1] = 1]$

Consistency: $[B_{i-1}[1] = B_i[0]]$

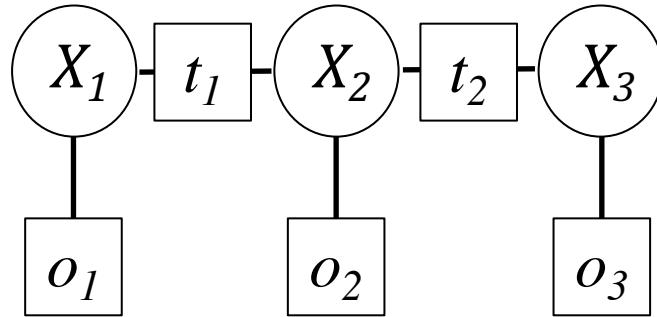
- Problem Modeling
- N-ary Constraints
- **Elimination Example**

Person Tracking Example



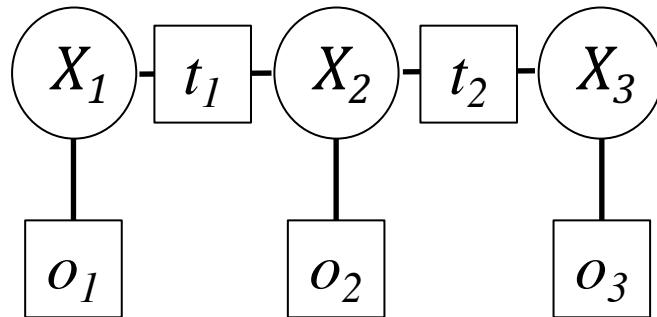
- Variables X_i : Location of object at position i

Person Tracking Example



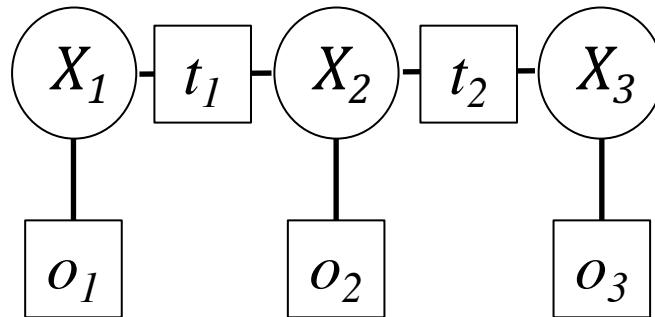
- Variables X_i : Location of object at position i
- Transition Factors $t_i(x_i, x_{i+1})$: object positions can't change too much

Person Tracking Example



- Variables X_i : Location of object at position i
- Transition Factors $t_i(x_i, x_{i+1})$: object positions can't change too much
- Observation Factors $o_i(x_i)$: noisy information compatible with position

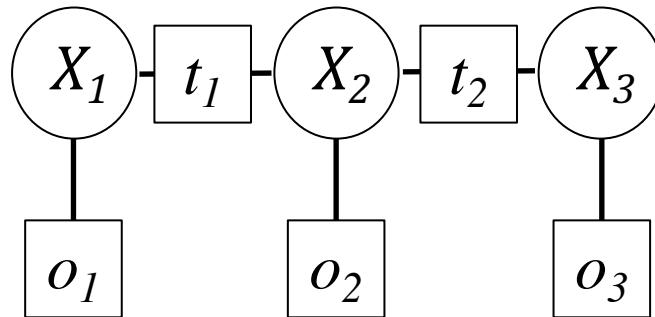
Person Tracking Example



- Variables X_i : Location of object at position i
- Transition Factors $t_i(x_i, x_{i+1})$: object positions can't change too much
- Observation Factors $o_i(x_i)$: noisy information compatible with position

```
def t(x, y):  
    if x == y: return 2  
    if abs(x - y) == 1: return 1  
    return 0
```

Person Tracking Example



- Variables X_i : Location of object at position i
- Transition Factors $t_i(x_i, x_{i+1})$: object positions can't change too much
- Observation Factors $o_i(x_i)$: noisy information compatible with position

```
def t(x, y):  
    if x == y: return 2  
    if abs(x - y) == 1: return 1  
    return 0
```

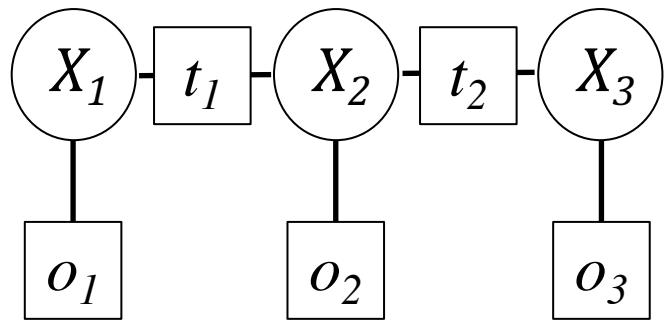
```
def o1(x): return t(x, 0)  
def o2(x): return t(x, 2)  
def o3(x): return t(x, 2)
```

Variable Elimination

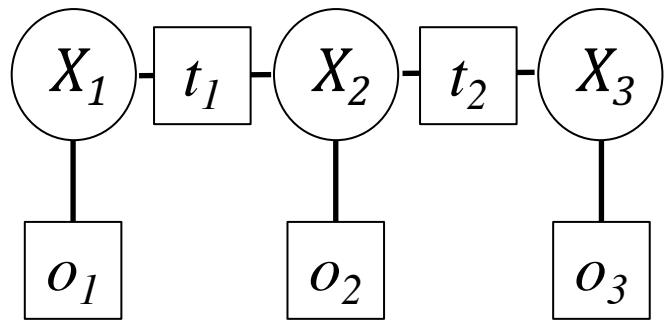
Definition: Elimination

- To **eliminate** a variable X_i , consider all factors f_1, \dots, f_k , that depend on X_i
- Remove X_i and f_1, \dots, f_k
- Add
$$f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$$

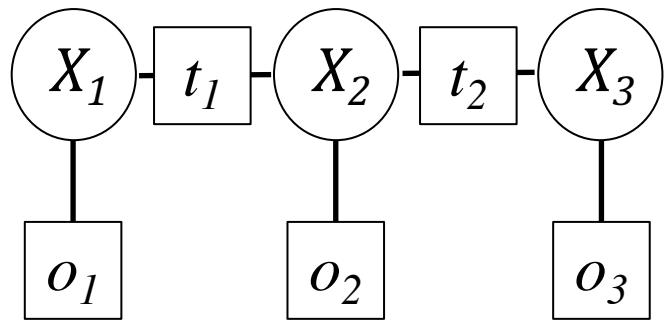
- Eliminate X_1



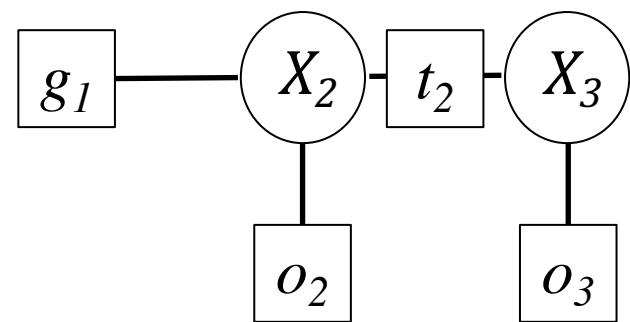
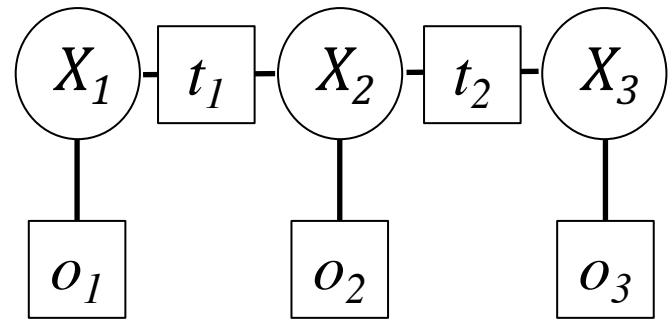
- Eliminate X_1
- Factors that depend on X_1 :
 - o_1, t_1



- Eliminate X_1
- Factors that depend on X_1 :
 - o_1, t_1
- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$



- Eliminate X_1
- Factors that depend on X_1 :
 - o_1, t_1
- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$
- $g_1(x_2) = \max_{x_1 \in \{0,1,2\}} o_1(x_1) \cdot t_1(x_1, x_2)$



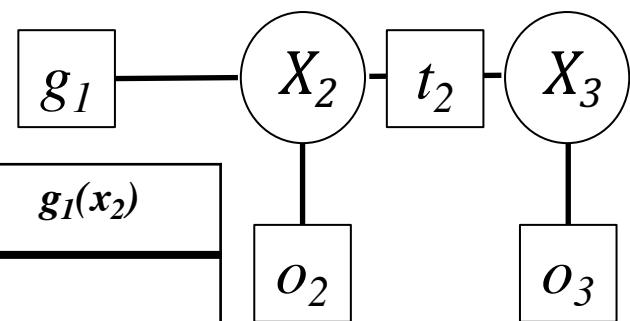
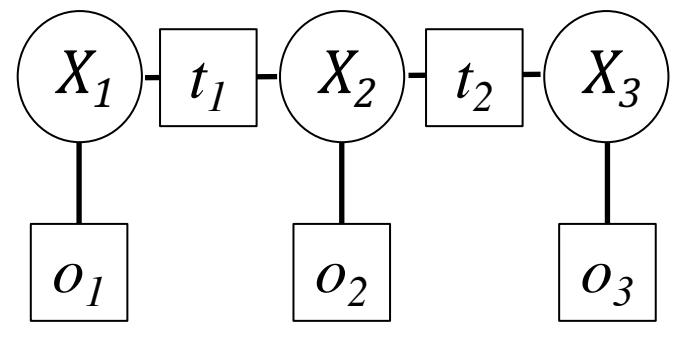
- Eliminate X_1
- Factors that depend on X_1 :

- o_1, t_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_1(x_2) = \max_{x_1 \in \{0,1,2\}} o_1(x_1) \cdot t_1(x_1, x_2)$

x_2	x_1	$o_1(x_1)$	$t_1(x_1, x_2)$	$o_1(x_1) \cdot t_1(x_1, x_2)$	$g_1(x_2)$
0	0	0	0	0	0
0	1				
0	2				
1	0	1	1	1	1
1	1				
1	2				
2	0	2	2	2	2
2	1				
2	2				



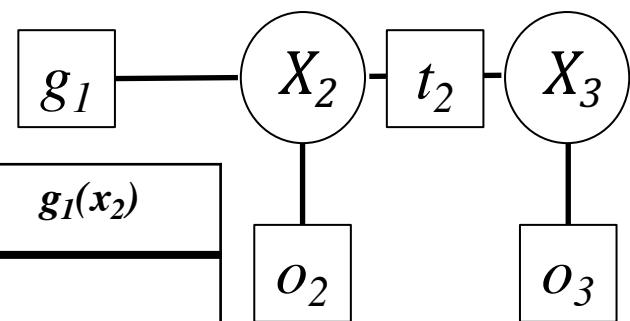
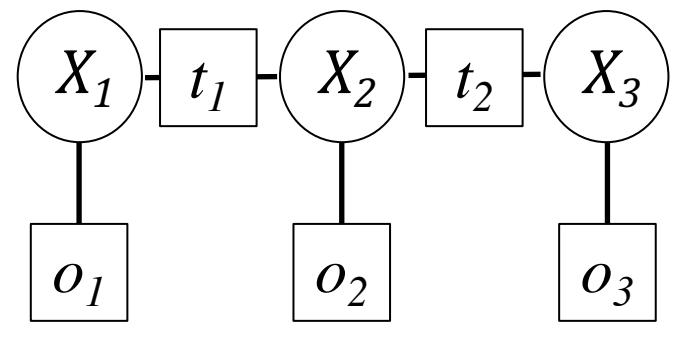
- Eliminate X_1
- Factors that depend on X_1 :

- o_1, t_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_1(x_2) = \max_{x_1 \in \{0,1,2\}} o_1(x_1) \cdot t_1(x_1, x_2)$

x_2	x_1	$o_1(x_1)$	$t_1(x_1, x_2)$	$o_1(x_1) \cdot t_1(x_1, x_2)$	$g_1(x_2)$
0	0	2			
0	1	1			
0	2	0			
1	0	2			
1	1	1			
1	2	0			
2	0	2			
2	1	1			
2	2	0			



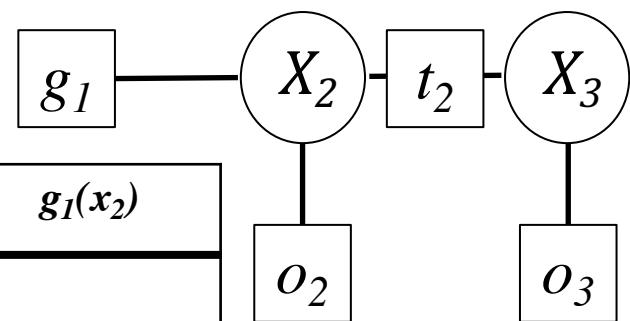
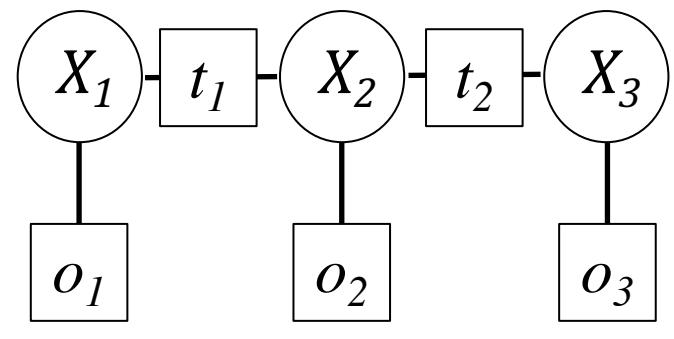
- Eliminate X_1
- Factors that depend on X_1 :

- o_1, t_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_1(x_2) = \max_{x_1 \in \{0,1,2\}} o_1(x_1) \cdot t_1(x_1, x_2)$

x_2	x_1	$o_1(x_1)$	$t_1(x_1, x_2)$	$o_1(x_1) \cdot t_1(x_1, x_2)$	$g_1(x_2)$
0	0	2	2		
0	1	1	1		
0	2	0	0		
1	0	2	1		
1	1	1	2		
1	2	0	1		
2	0	2	0		
2	1	1	1		
2	2	0	2		



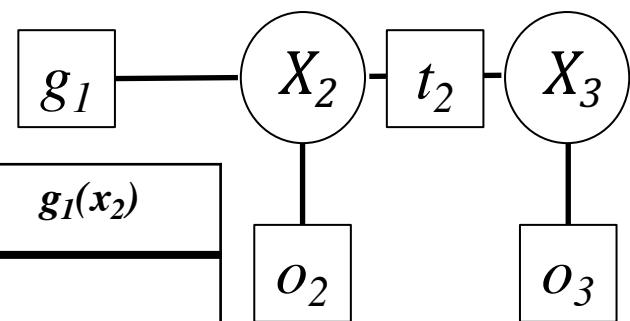
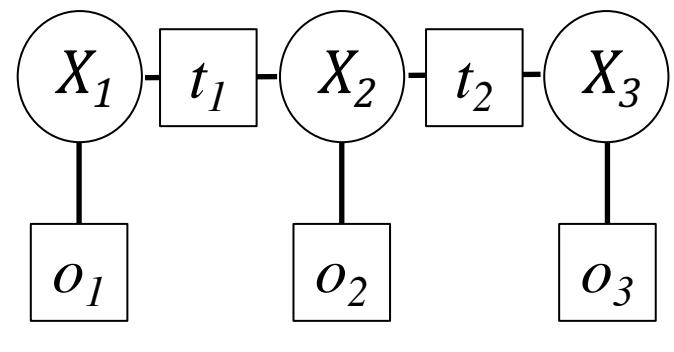
- Eliminate X_1
- Factors that depend on X_1 :

- o_1, t_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_1(x_2) = \max_{x_1 \in \{0,1,2\}} o_1(x_1) \cdot t_1(x_1, x_2)$

x_2	x_1	$o_1(x_1)$	$t_1(x_1, x_2)$	$o_1(x_1) \cdot t_1(x_1, x_2)$	$g_1(x_2)$
0	0	2	2	4	
0	1	1	1	1	
0	2	0	0	0	
1	0	2	1	2	
1	1	1	2	2	
1	2	0	1	0	
2	0	2	0	0	
2	1	1	1	1	
2	2	0	2	0	



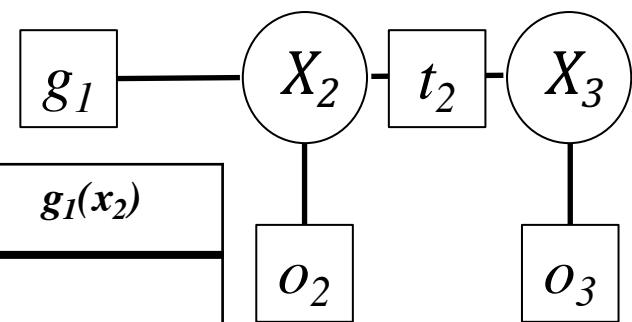
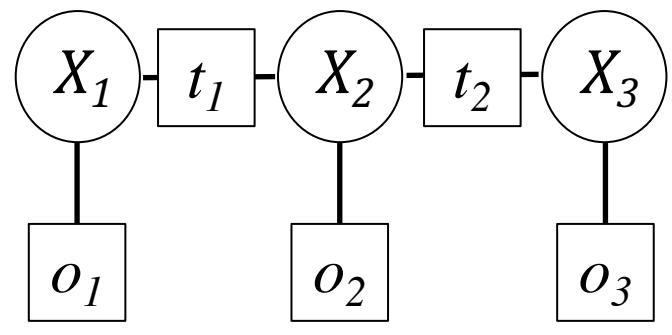
- Eliminate X_I
- Factors that depend on X_I :

- o_I, t_I

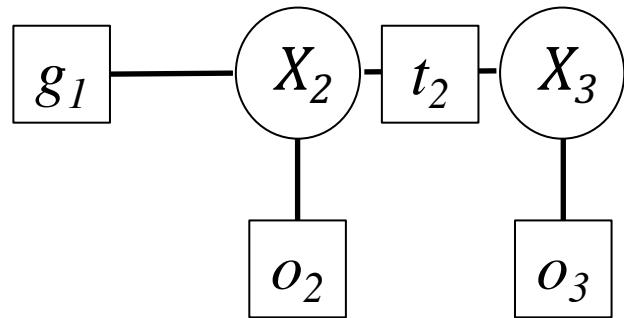
- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_1(x_2) = \max_{x_1 \in \{0,1,2\}} o_1(x_1) \cdot t_1(x_1, x_2)$

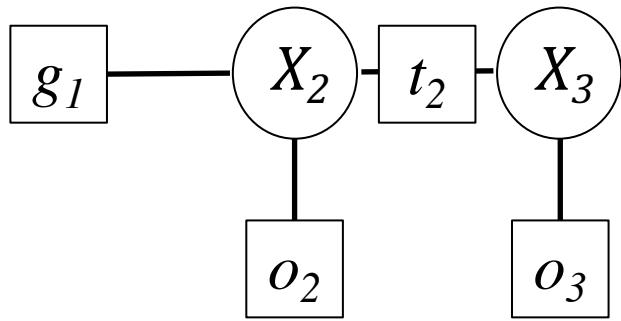
x_2	x_I	$o_I(x_I)$	$t_I(x_I, x_2)$	$o_I(x_I) \cdot t_I(x_I, x_2)$	$g_1(x_2)$
0	0	2	2	4	4: $\{x_I: 0\}$
0	1	1	1	1	
0	2	0	0	0	
1	0	2	1	2	2: $\{x_I: 1\}$
1	1	1	2	2	
1	2	0	1	0	
2	0	2	0	0	1: $\{x_I: 1\}$
2	1	1	1	1	
2	2	0	2	0	



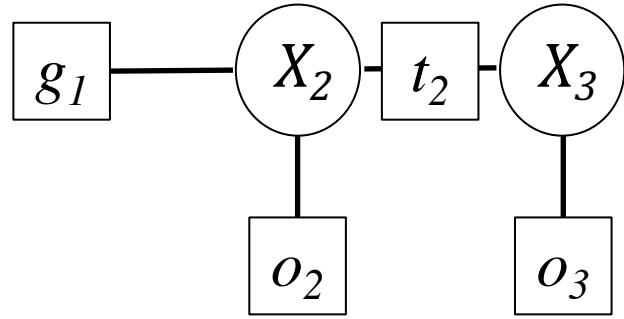
- Eliminate X_2



- Eliminate X_2
- Factors that depend on X_2 :
 - o_2, t_2, g_1



- Eliminate X_2
- Factors that depend on X_2 :
 - o_2, t_2, g_1
- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

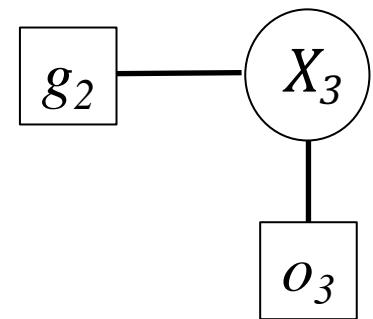
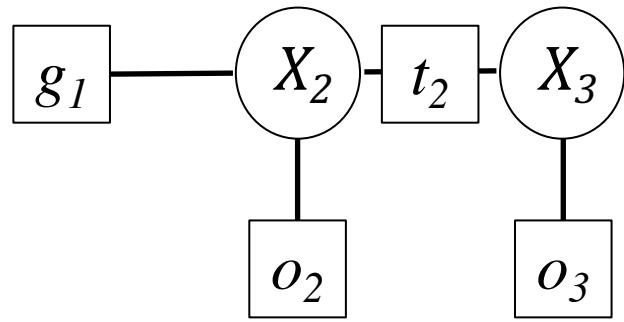


- Eliminate X_2
- Factors that depend on X_2 :

- o_2, t_2, g_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_2(x_3) = \max_{x_2 \in \{0,1,2\}} g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$

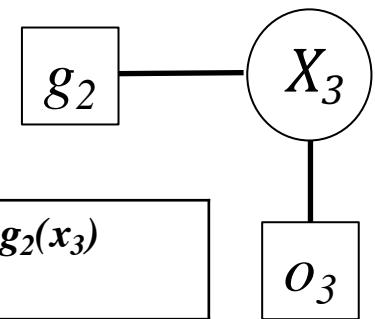
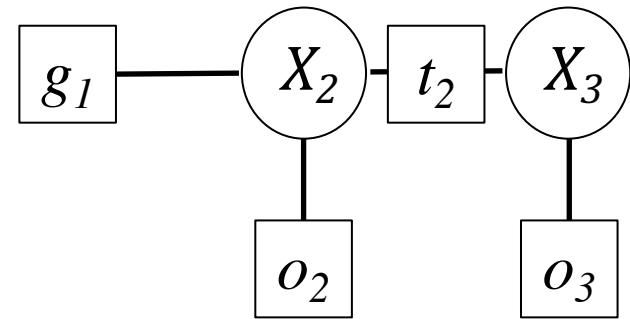


- Eliminate X_2
- Factors that depend on X_2 :

- o_2, t_2, g_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_2(x_3) = \max_{x_2 \in \{0,1,2\}} g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$



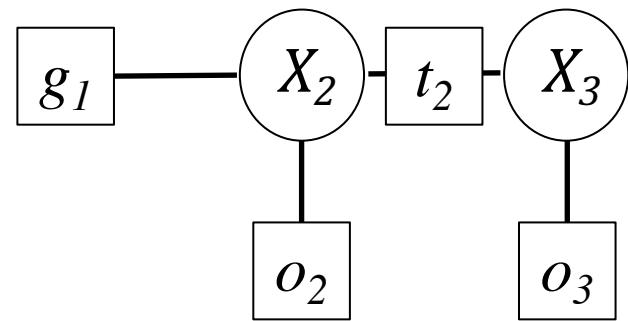
x_3	x_2	$g_1(x_2)$	$o_2(x_2)$	$t_2(x_2, x_3)$	$g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$	$g_2(x_3)$
0	0					
0	1					
0	2					
1	0					
1	1					
1	2					
2	0					
2	1					
2	2					

- Eliminate X_2
- Factors that depend on X_2 :

- o_2, t_2, g_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_2(x_3) = \max_{x_2 \in \{0,1,2\}} g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$



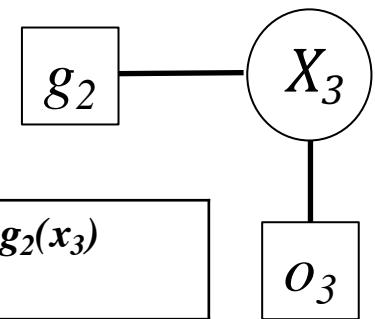
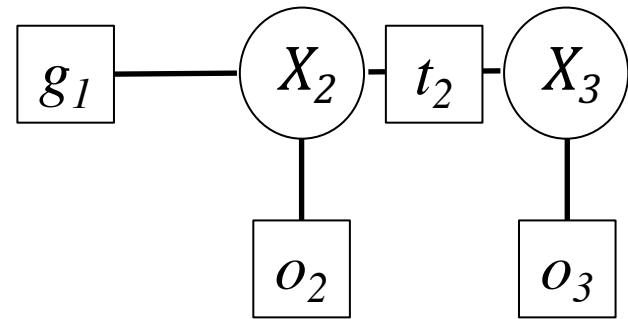
x_3	x_2	$g_1(x_2)$	$o_2(x_2)$	$t_2(x_2, x_3)$	$g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$	$g_2(x_3)$
0	0	4: $\{x_I: 0\}$				
0	1	2: $\{x_I: I\}$				
0	2	1: $\{x_I: I\}$				
1	0	4: $\{x_I: 0\}$				
1	1	2: $\{x_I: I\}$				
1	2	1: $\{x_I: I\}$				
2	0	4: $\{x_I: 0\}$				
2	1	2: $\{x_I: I\}$				
2	2	1: $\{x_I: I\}$				

- Eliminate X_2
- Factors that depend on X_2 :

- o_2, t_2, g_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_2(x_3) = \max_{x_2 \in \{0,1,2\}} g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$



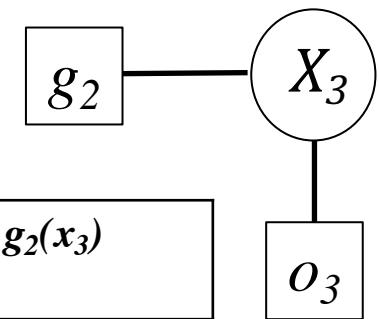
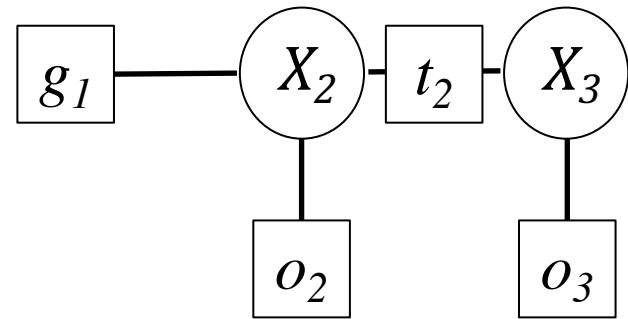
x_3	x_2	$g_1(x_2)$	$o_2(x_2)$	$t_2(x_2, x_3)$	$g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$	$g_2(x_3)$
0	0	4: $\{x_I: 0\}$	0			
0	1	2: $\{x_I: I\}$	1			
0	2	1: $\{x_I: I\}$	2			
1	0	4: $\{x_I: 0\}$	0			
1	1	2: $\{x_I: I\}$	1			
1	2	1: $\{x_I: I\}$	2			
2	0	4: $\{x_I: 0\}$	0			
2	1	2: $\{x_I: I\}$	1			
2	2	1: $\{x_I: I\}$	2			

- Eliminate X_2
- Factors that depend on X_2 :

- o_2, t_2, g_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_2(x_3) = \max_{x_2 \in \{0,1,2\}} g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$



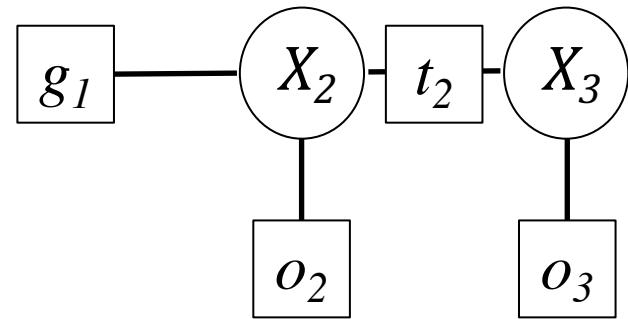
x_3	x_2	$g_1(x_2)$	$o_2(x_2)$	$t_2(x_2, x_3)$	$g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$	$g_2(x_3)$
0	0	4: $\{x_I: 0\}$	0	2		
0	1	2: $\{x_I: I\}$	1	1		
0	2	1: $\{x_I: I\}$	2	0		
1	0	4: $\{x_I: 0\}$	0	1		
1	1	2: $\{x_I: I\}$	1	2		
1	2	1: $\{x_I: I\}$	2	1		
2	0	4: $\{x_I: 0\}$	0	0		
2	1	2: $\{x_I: I\}$	1	1		
2	2	1: $\{x_I: I\}$	2	2		

- Eliminate X_2
- Factors that depend on X_2 :

- o_2, t_2, g_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

- $g_2(x_3) = \max_{x_2 \in \{0,1,2\}} g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$



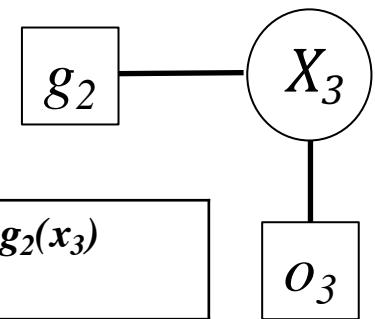
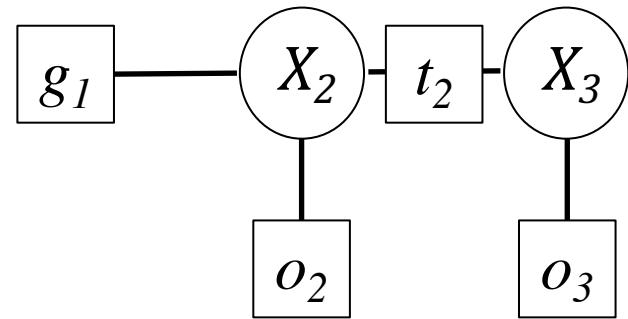
x_3	x_2	$g_1(x_2)$	$o_2(x_2)$	$t_2(x_2, x_3)$	$g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$	$g_2(x_3)$
0	0	4: $\{x_I: 0\}$	0	2	0	
0	1	2: $\{x_I: I\}$	1	1	2	
0	2	1: $\{x_I: I\}$	2	0	2	
1	0	4: $\{x_I: 0\}$	0	1	4	
1	1	2: $\{x_I: I\}$	1	2	4	
1	2	1: $\{x_I: I\}$	2	1	2	
2	0	4: $\{x_I: 0\}$	0	0	0	
2	1	2: $\{x_I: I\}$	1	1	2	
2	2	1: $\{x_I: I\}$	2	2	4	

- Eliminate X_2
- Factors that depend on X_2 :

- o_2, t_2, g_1

- Add $f_{new}(x) = \max_{x_i} \prod_{j=1}^k f_j(x)$

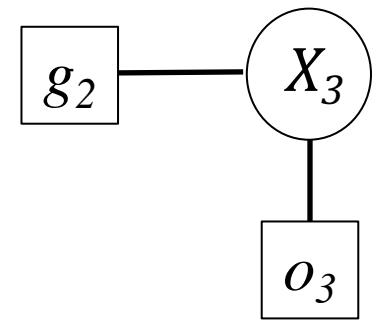
- $g_2(x_3) = \max_{x_2 \in \{0,1,2\}} g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$



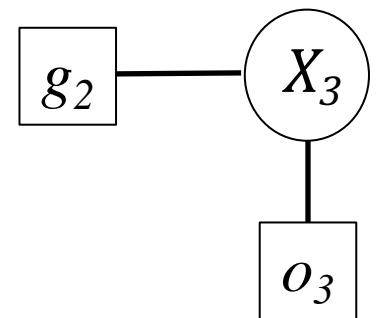
x_3	x_2	$g_1(x_2)$	$o_2(x_2)$	$t_2(x_2, x_3)$	$g_1(x_2) \cdot o_2(x_2) \cdot t_2(x_2, x_3)$	$g_2(x_3)$
0	0	4: $\{x_1: 0\}$	0	2	0	2: $\{x_1: 1, x_2: 2\}$
0	1	2: $\{x_1: 1\}$	1	1	2	
0	2	1: $\{x_1: 1\}$	2	0	2	
1	0	4: $\{x_1: 0\}$	0	1	4	4: $\{x_1: 1, x_2: 1\}$
1	1	2: $\{x_1: 1\}$	1	2	4	
1	2	1: $\{x_1: 1\}$	2	1	2	
2	0	4: $\{x_1: 0\}$	0	0	0	4: $\{x_1: 1, x_2: 2\}$
2	1	2: $\{x_1: 1\}$	1	1	2	
2	2	1: $\{x_1: 1\}$	2	2	4	

- We are left with:

$$\max_{x_3 \in \{0,1,2\}} g_2(x_3) \cdot o_3(x_3)$$



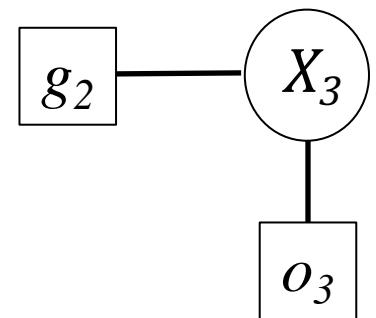
- We are left with:



$$\max_{x_3 \in \{0,1,2\}} g_2(x_3) \cdot o_3(x_3)$$

x_3	$g_2(x_3)$	$o_3(x_3)$	$g_2(x_3) \cdot o_3(x_3)$	<i>Optimal Weight</i>
0				
1				
2				

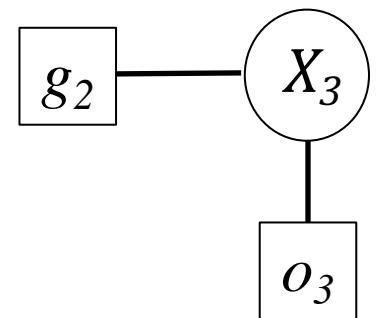
- We are left with:



$$\max_{x_3 \in \{0,1,2\}} g_2(x_3) \cdot o_3(x_3)$$

x_3	$g_2(x_3)$	$o_3(x_3)$	$g_2(x_3) o_3(x_3)$	<i>Optimal Weight</i>
0	2: $\{x_1 : 1, x_2 : 2\}$	0		
1	4: $\{x_1 : 1, x_2 : 1\}$	1		
2	4: $\{x_1 : 1, x_2 : 2\}$	2		

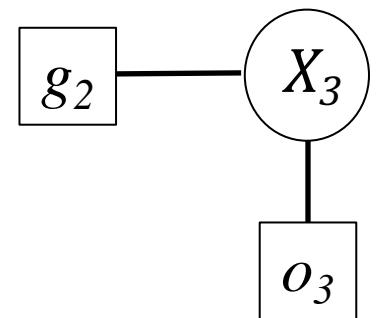
- We are left with:



$$\max_{x_3 \in \{0,1,2\}} g_2(x_3) \cdot o_3(x_3)$$

x_3	$g_2(x_3)$	$o_3(x_3)$	$g_2(x_3) o_3(x_3)$	<i>Optimal Weight</i>
0	2: $\{x_1 : 1, x_2 : 2\}$	0	2	
1	4: $\{x_1 : 1, x_2 : 1\}$	1	4	
2	4: $\{x_1 : 1, x_2 : 2\}$	2	8	

- We are left with:



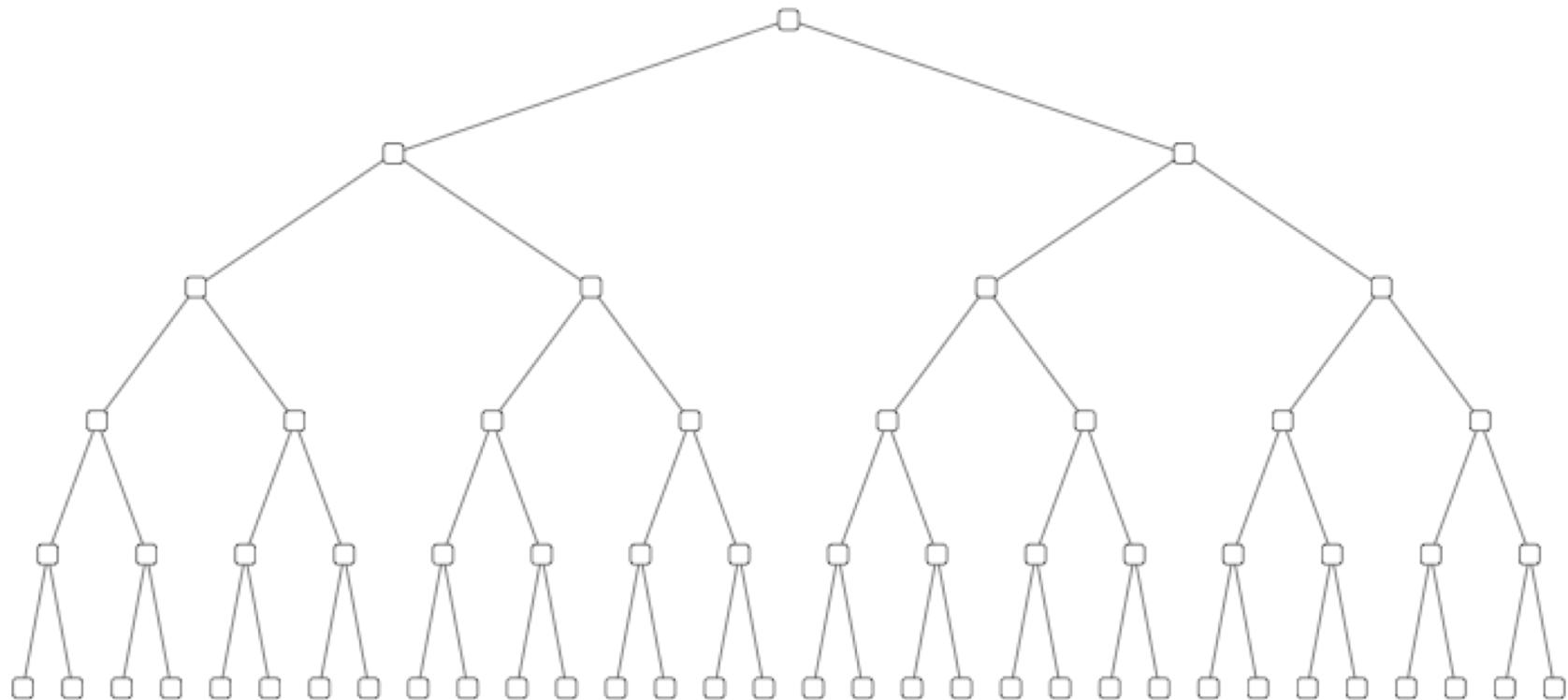
$$\max_{x_3 \in \{0,1,2\}} g_2(x_3) \cdot o_3(x_3)$$

x_3	$g_2(x_3)$	$o_3(x_3)$	$g_2(x_3) \cdot o_3(x_3)$	<i>Optimal Weight</i>
0	2: $\{x_1 : 1, x_2 : 2\}$	0	2	8: $\{x_1 : 1, x_2 : 2, x_3 : 2\}$
1	4: $\{x_1 : 1, x_2 : 1\}$	1	4	
2	4: $\{x_1 : 1, x_2 : 2\}$	2	8	

Search Techniques

- Backtracking
- Beam Search
- Gibbs Sampling
- Conditioning
- Elimination

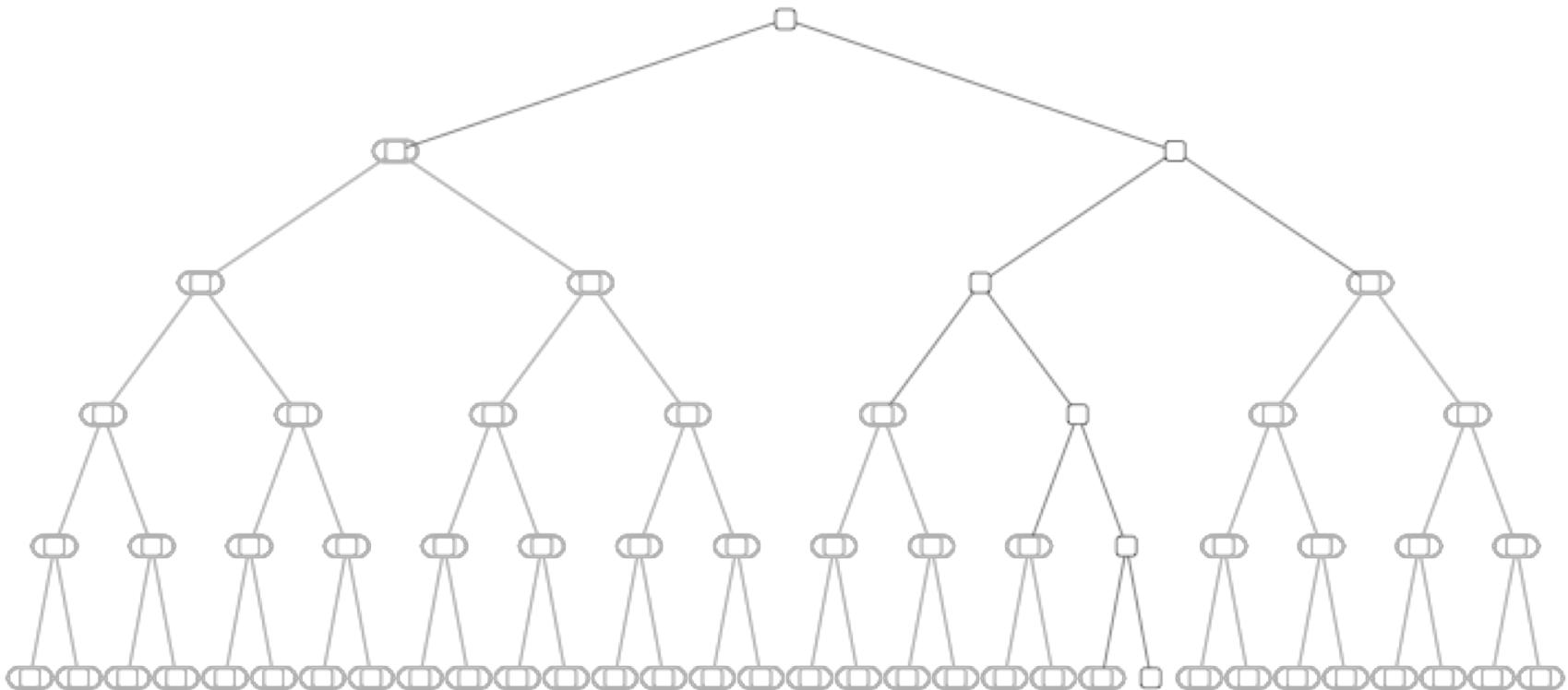
Backtracking search



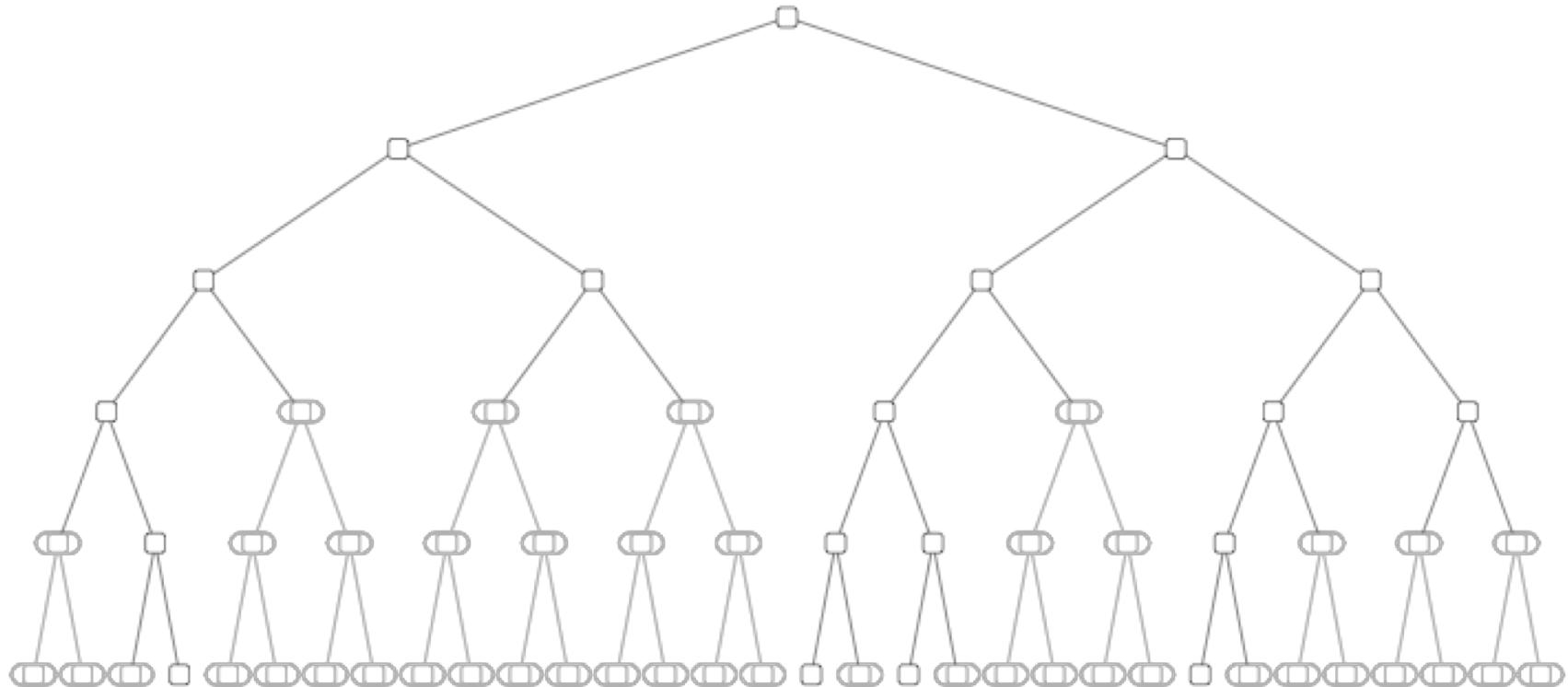
Search Techniques

- Backtracking
- Beam Search
- Gibbs Sampling
- Conditioning
- Elimination

Greedy search



Beam search



Beam size $K = 4$

Search Techniques

- Backtracking
- Beam Search
- Iterated Conditional Modes
- **Gibbs Sampling**
- Conditioning
- Elimination

Gibbs sampling

Sometimes, need to go downhill to go uphill...



Key idea: randomness

Sample an assignment with probability proportional to its weight.



Example: Gibbs sampling

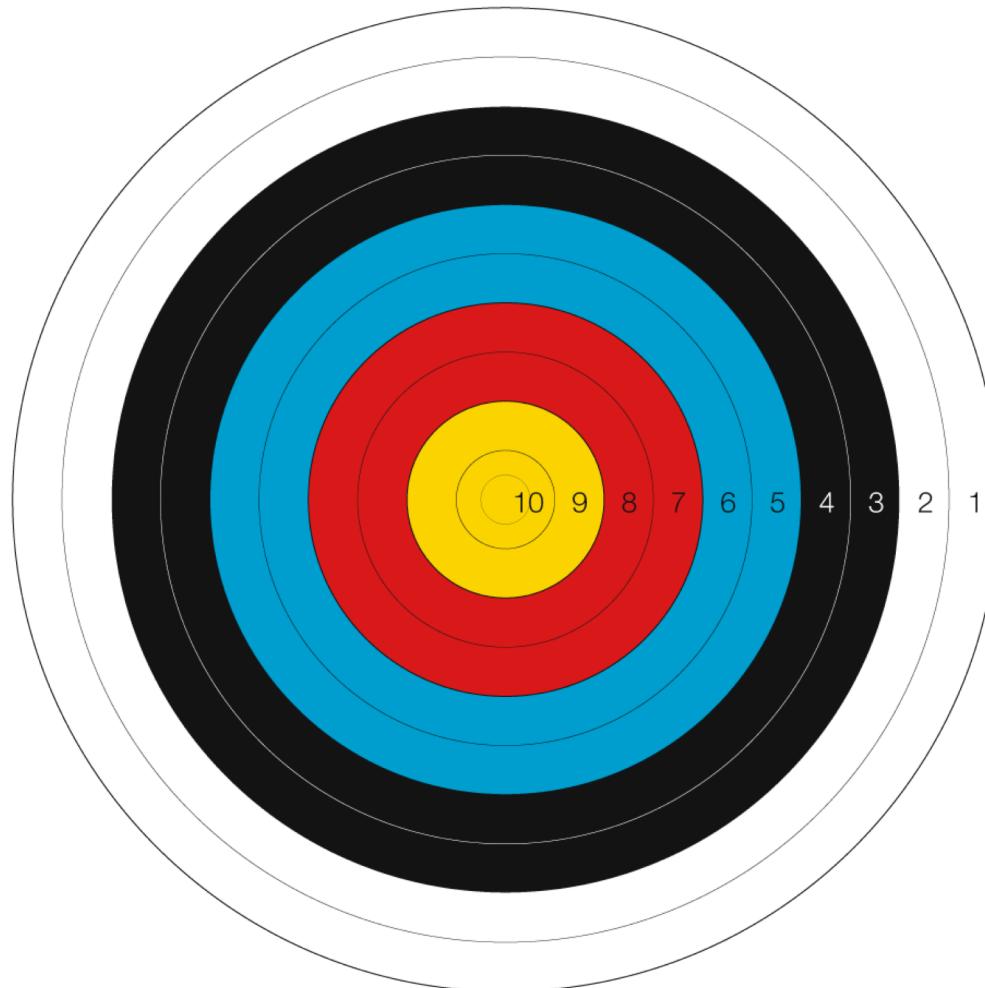
Weight($x \cup \{X_2 : 0\}\} = 1$ prob. 0.2

Weight($x \cup \{X_2 : 1\}\} = 2$ prob. 0.4

Weight($x \cup \{X_2 : 2\}\} = 2$ prob. 0.4



Use Randomness

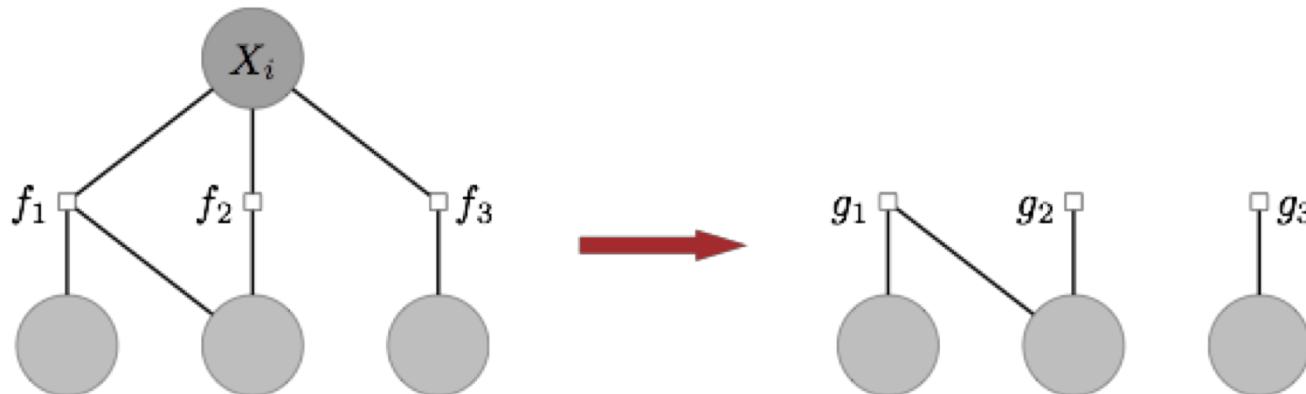


Search Techniques

- Backtracking
- Beam Search
- ICM
- Gibbs Sampling
- **Conditioning**
- Elimination

Conditioning: general

Graphically: remove edges from X_i to dependent factors



Definition: conditioning

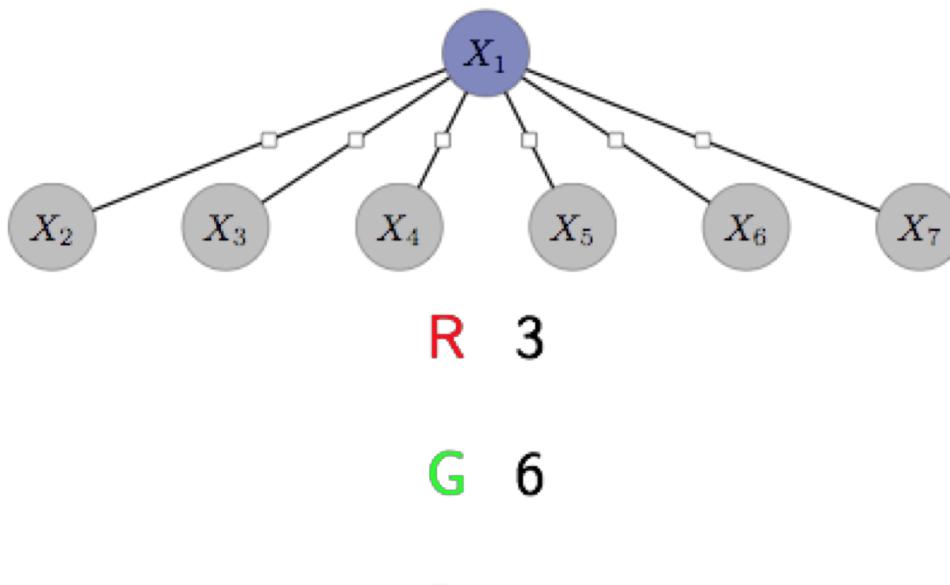
- To **condition** on a variable $X_i = v$, consider all factors f_1, \dots, f_k that depend on X_i .
- Remove X_i and f_1, \dots, f_k .
- Add $g_j(x) = f_j(x \cup \{X_i : v\})$ for $j = 1, \dots, k$.

Using conditional independence

For each value $v = \text{R, G, B}$:

Condition on $X_1 = v$.

Find the maximum weight assignment (easy).



maximum weight is 6

Search Techniques

- Backtracking
- Beam Search
- ICM
- Gibbs Sampling
- Conditioning
- Elimination