Constraint Satisfaction Problems (CSPs)

CS 221 Section – 10/31/19

Chuma Kabaghe Will Deaderick

Agenda

- CSP Problem Modeling
- N-ary Constraints
- Exam Problem Solving

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.)

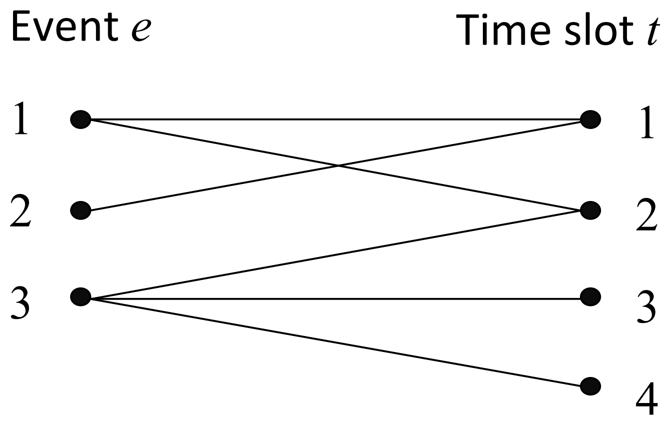
Agenda

- CSP Problem Modeling
- N-ary Constraints
- Exam Problem Solving

Event Scheduling

Setup:

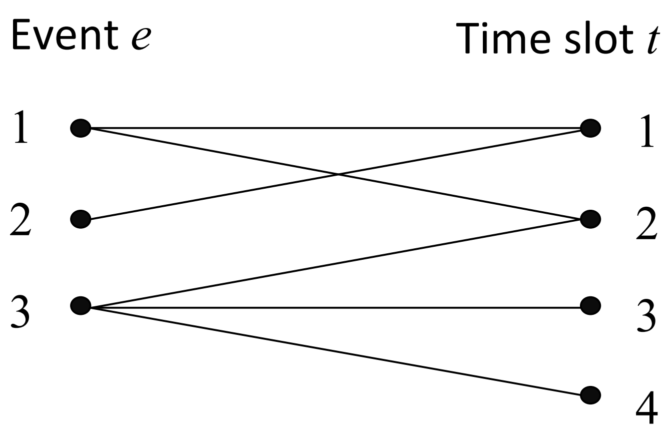
- Have E events and T time slots
- Each event e must be put in **exactly one** time slot
- Each time slot t can have **at most one** event
- Event e only allowed at time slot t if (e, t) in A



Event Scheduling

Setup:

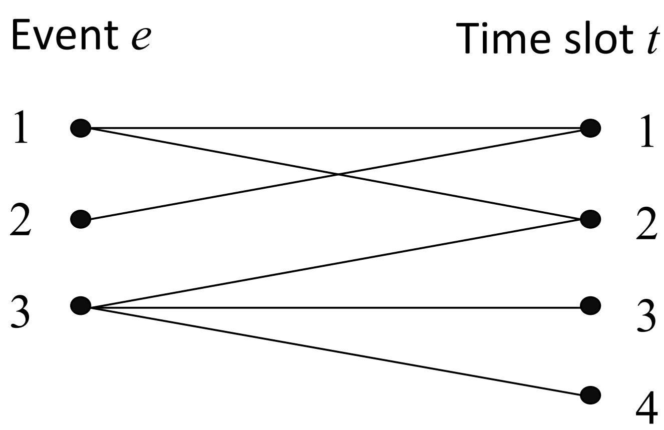
- Have E events and T time slots
- Each event e must be put in **exactly one** time slot
- Each time slot t can have **at most one** event
- Event e only allowed at time slot t if (e, t) in A



Event Scheduling

Formulation 1a:

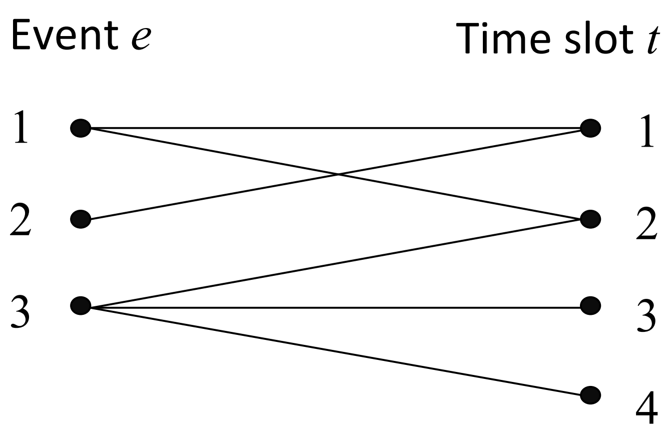
- Variables for each event e, $X_e \in \{1,\ldots,T\}$



Event Scheduling

Formulation 1a:

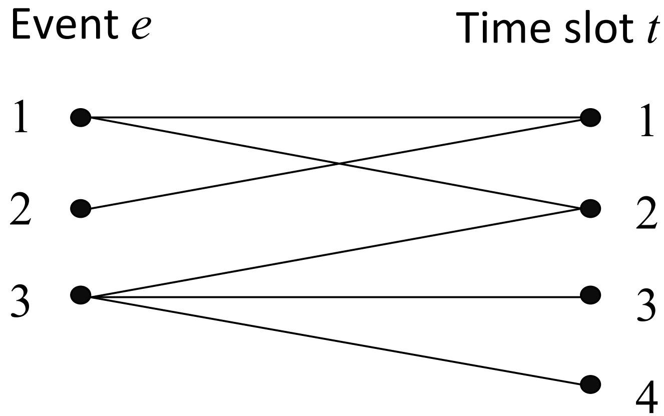
- Variables for each event e, $X_e \in \{1,\ldots,T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$



Event Scheduling

Formulation 1a:

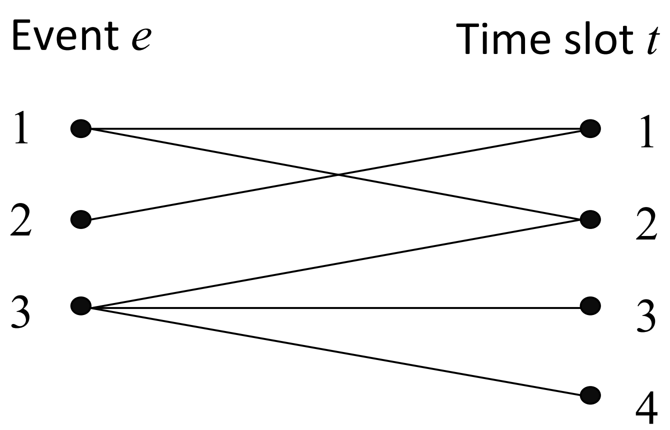
- Variables for each event e, $X_e \in \{1, \ldots, T\}$
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$
- Constraints (only schedule allowed times): for each event e, enforce $[(e, X_e) \in A]$



Event Scheduling

Formulation 1a:

- Variables for each event e, $X_e \in \{1, \ldots, T\}$
- 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 $[(e, X_e) \in A]$

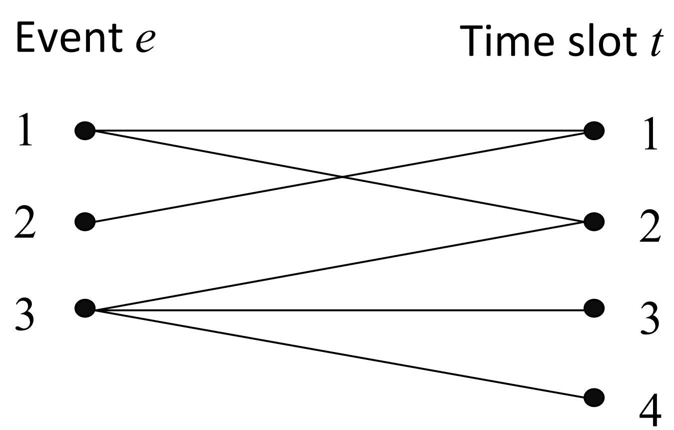


$\{X_1 : 1, X_2 : 1, X_3 : 3\}$

Event Scheduling

Formulation 1a:

- Variables for each event e, $X_e \in \{1, \ldots, T\}$
- 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 $[(e, X_e) \in A]$



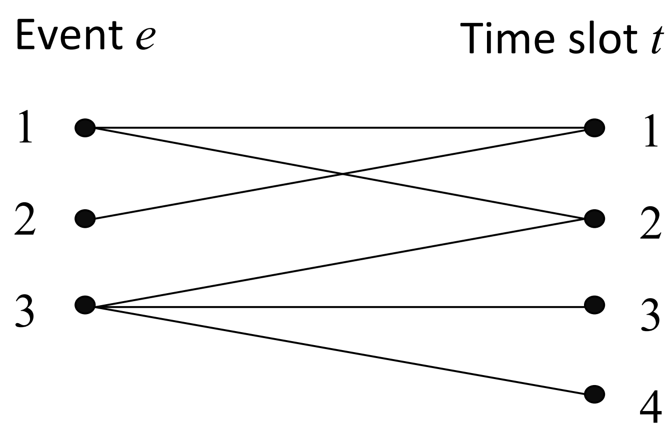
$\{X_1 : 1, X_2 : 1, X_3 : 3\}$

Bad! ($X_1 = X_2$)

Event Scheduling

Formulation 1a:

- Variables for each event e, $X_e \in \{1,\ldots,T\}$
- 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 $[(e, X_e) \in A]$

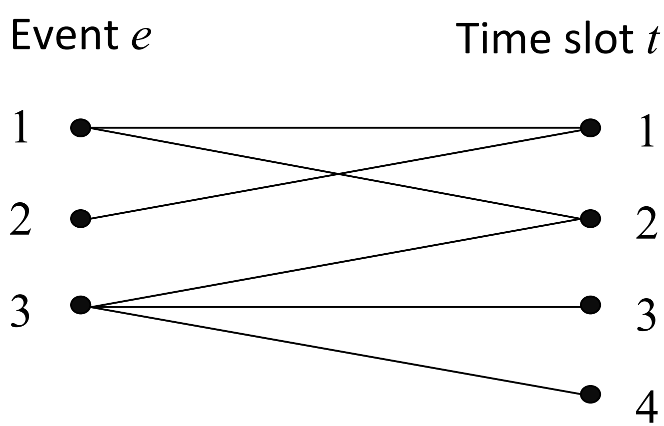


$\{X_1 : 1, X_2 : 4, X_3 : 3\}$

Event Scheduling

Formulation 1a:

- Variables for each event e, $X_e \in \{1,\ldots,T\}$
- 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 $[(e, X_e) \in A]$



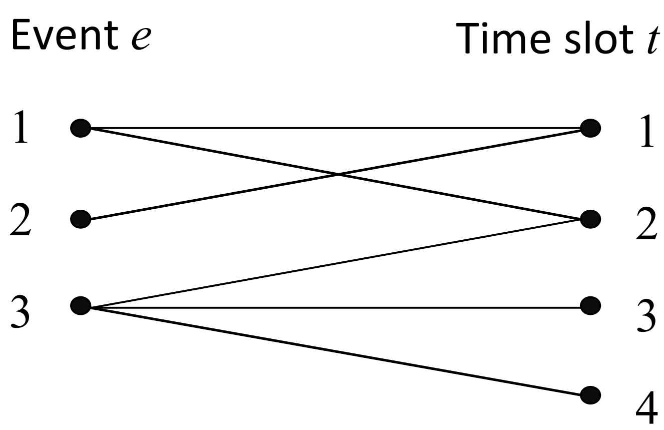
$\{X_1 : 1, X_2 : 4, X_3 : 3\}$

Bad! ((2, 4) not in A)

Event Scheduling

Formulation 1a:

- Variables for each event e, $X_e \in \{1, \ldots, T\}$
- 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 $[(e, X_e) \in A]$

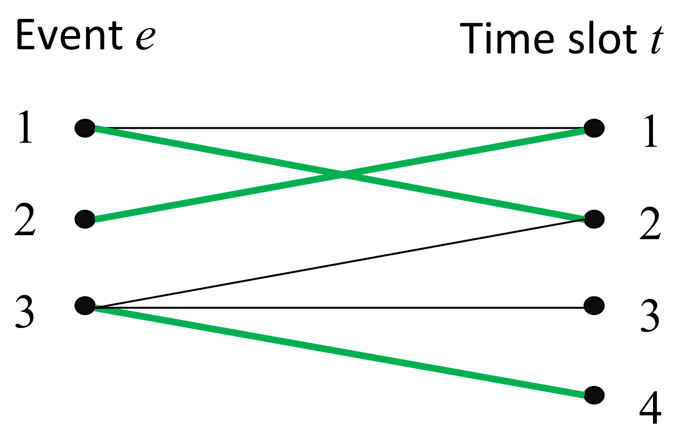


$\{X_1 : 2, X_2 : 1, X_3 : 4\}$

Event Scheduling

Formulation 1a:

- Variables for each event e, $X_e \in \{1,\ldots,T\}$
- 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 $[(e, X_e) \in A]$



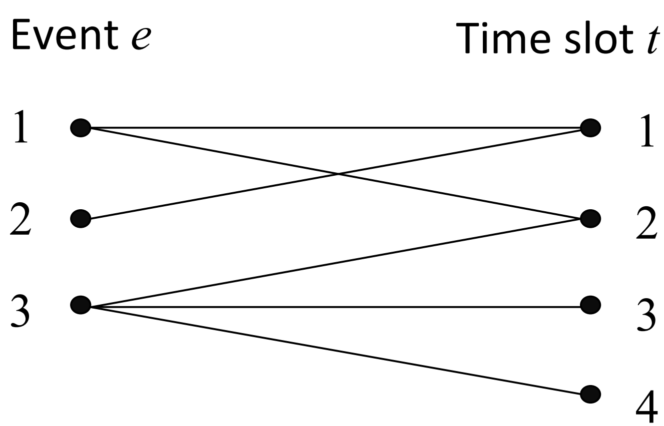
$\{X_1 : 2, X_2 : 1, X_3 : 4\}$

GOOD!

Event Scheduling

Formulation 1a:

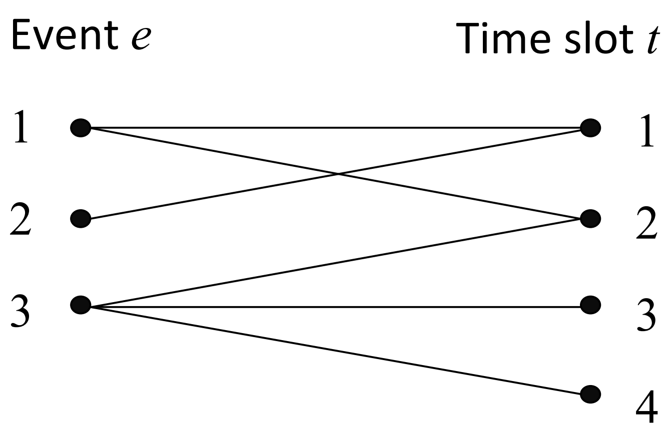
- Variables for each event *e*, $X_e \in \{1, \ldots, T\}$
- Constraints (only one event per time slot): for each pair of even $e \neq e'$, enforce $[X_e \neq X_{e'}]$
- Constraints (only schedule allowed times): for each event *e*, enforce $[(e, X_e) \in A]$



Event Scheduling

Formulation 1b:

- Variables for each event *e*, $X_1, \ldots, X_E$

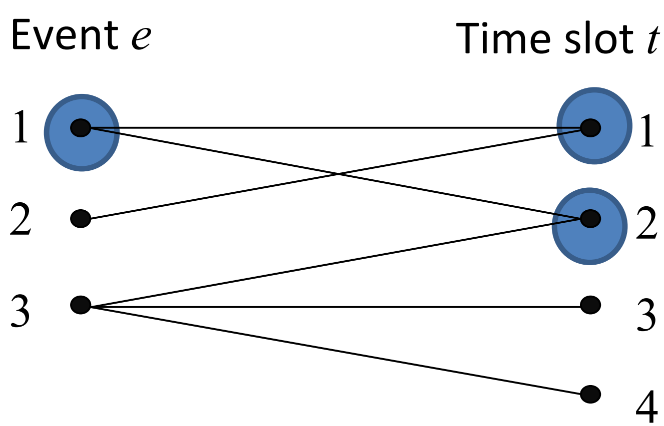


Event Scheduling

Formulation 1b:

- Variables for each event e, $X_1, \ldots, X_E$

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



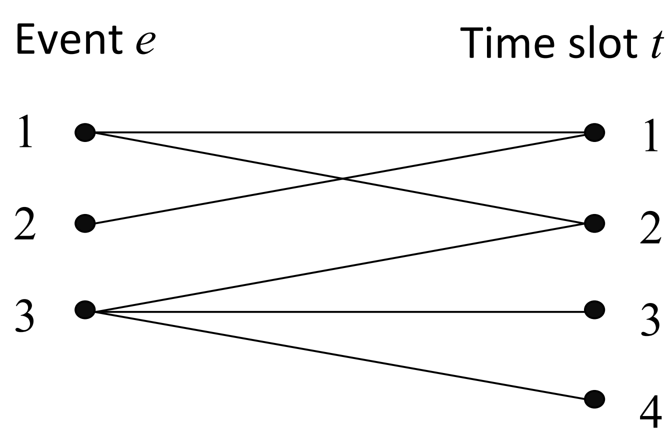
Event Scheduling

Formulation 1b:

- Variables for each event e, $X_1, \ldots, X_E$

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

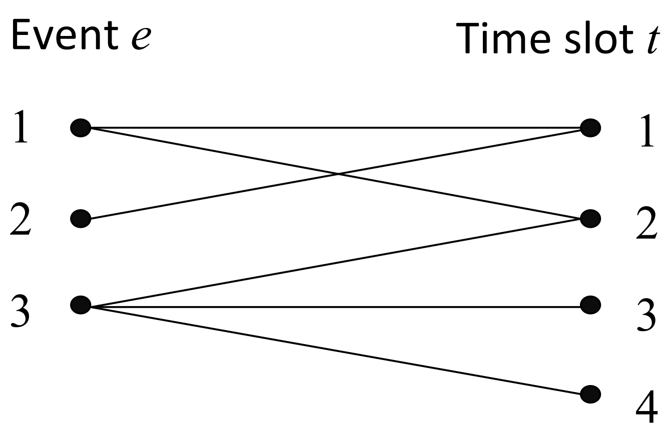
- Constraints (only one event per time slot): for each pair of events $e \neq e'$, enforce $[X_e \neq X_{e'}]$



Event Scheduling

Formulation 2a:

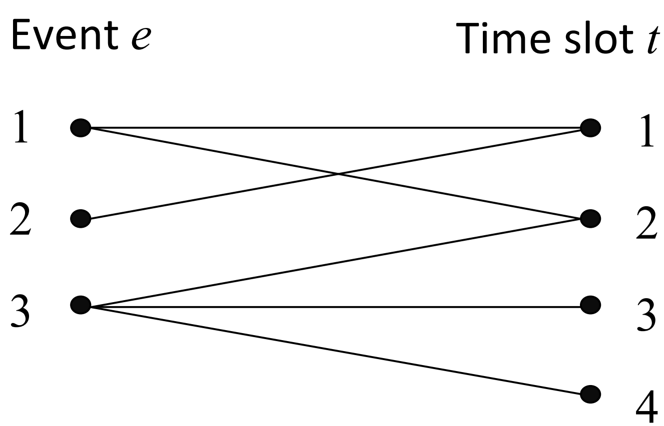
- Variables for each time slot t: $Y_t \in \{1,\ldots,E\} \cup \{\varnothing\}$



Event Scheduling

Formulation 2a:

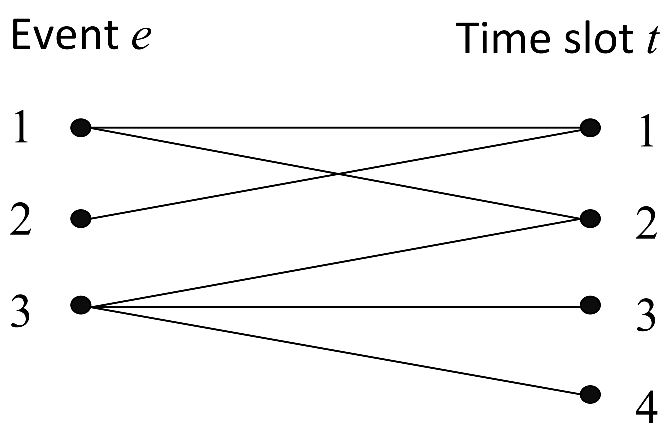
- Variables for each time slot t: $Y_t \in \{1,\ldots,E\} \cup \{\emptyset\}$
- Constraints (each event is scheduled exactly once): for each event e, enforce $[Y_t = e \text{ for exactly one } t]$



Event Scheduling

Formulation 2a:

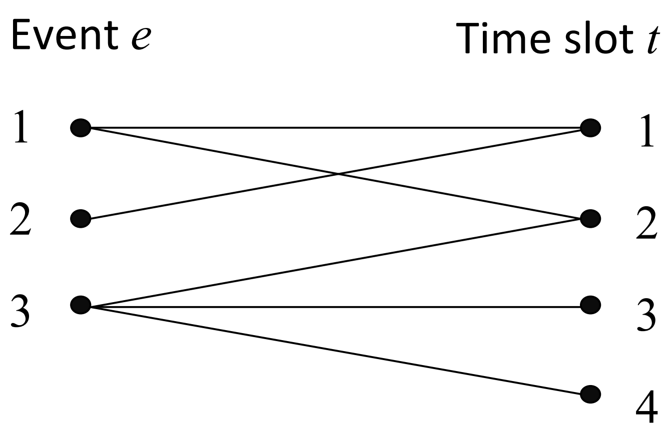
- Variables for each time slot t: $Y_t \in \{1,\ldots,E\} \cup \{\varnothing\}$
- Constraints (each event is scheduled exactly once): for each event e, enforce $[Y_t = e \text{ for exactly one } t]$
- Constraints (only schedule allowed times): for each time slot t, enforce $[Y_t = \varnothing \text{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



Event Scheduling

Formulation 2a:

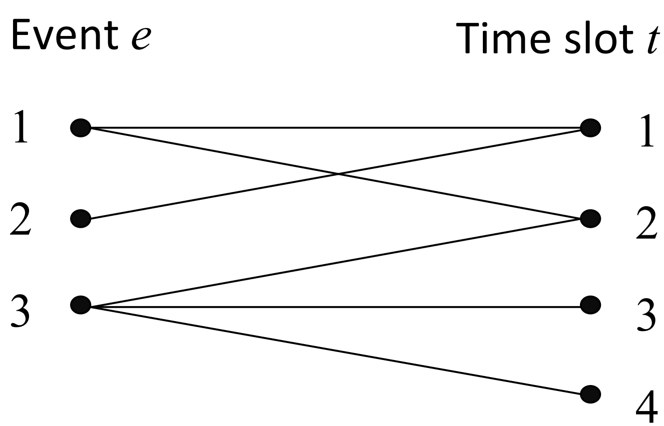
- Variables for each time slot t: $Y_t \in \{1,\ldots,E\} \cup \{\varnothing\}$
- Constraints (each event is scheduled exactly once): for each event e, enforce $[Y_t = e$ for exactly one $t]$
- Constraints (only schedule allowed times): for each time slot t, enforce $[Y_t = \varnothing$ or $(Y_t, t)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



Event Scheduling

Formulation 2b:

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

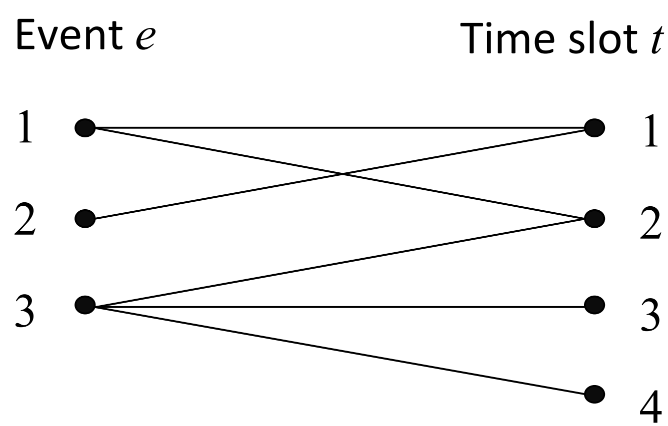


Event Scheduling

Formulation 2b:

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

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



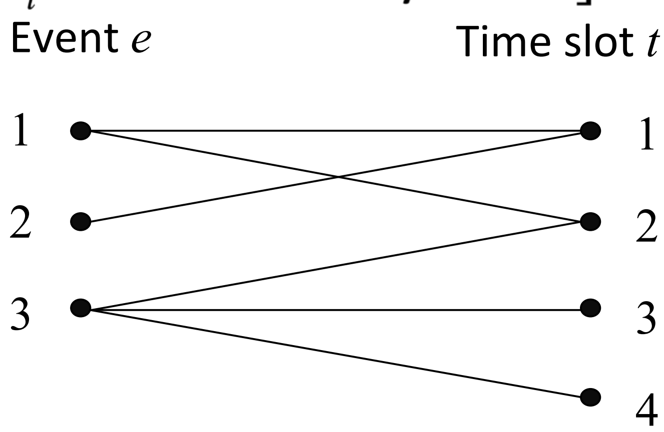
Event Scheduling

Formulation 2b:

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

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

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



Event Scheduling

Formulation 1a:

E variables with domain size T, and O(E^2) binary constraints.

- Variables for each event e, $X_e \in \{1,\ldots,T\}$
- 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 $[(e, X_e) \in A]$

T variables with domain size E+1
O(T^2) variables with domain size 2 and O(T^2) binary constraints.

Formulation 2a:

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

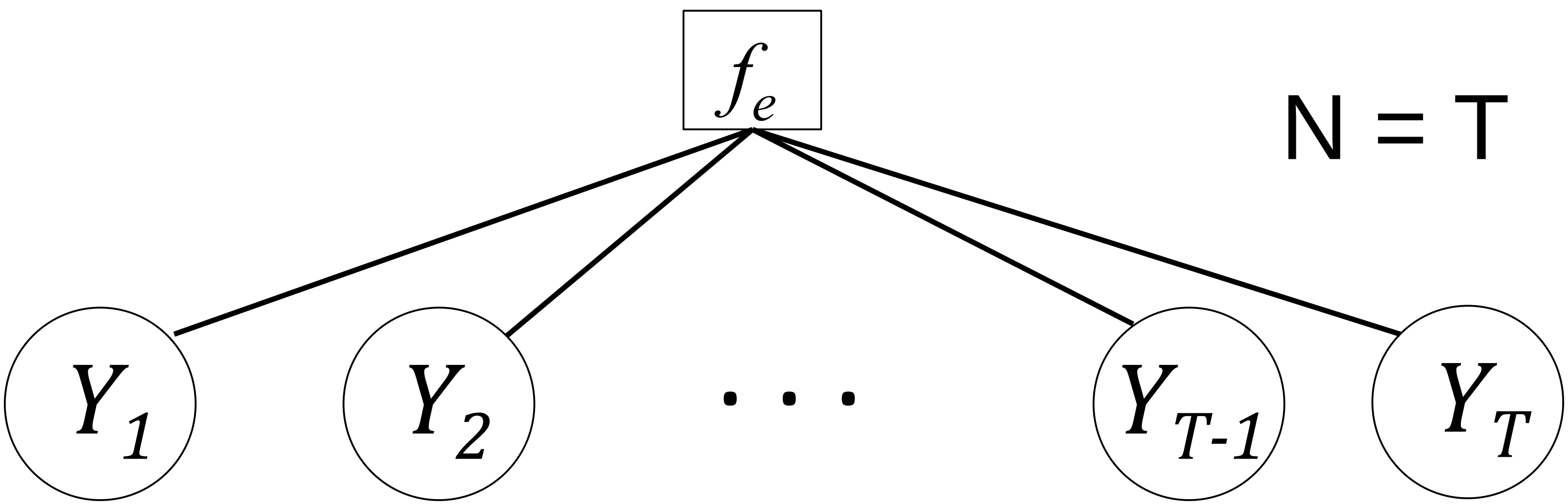
Agenda

- CSP Problem Modeling
- N-ary Constraints
- Exam Problem Solving

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.

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.

$e = 1$

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

Factors:

Initialization: $[A_0 = 0]$

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.

e = 1

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

Factors:

Initialization: $[A_0 = 0]$

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

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.

e = 1

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

Factors:

Initialization: $[A_0 = 0]$

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

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

$e = 1$

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

$e = 1$

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

$e = 1$

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

$e = 1$

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.

$e = 1$

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

Factors:

Initialization: $[A_0 = 0]$

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

Final Output: $1[A_T = 1]$

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.

$e = 1$

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

Factors:

Initialization: $[A_0 = 0]$
Processing: $[A_i = \min(A_{i-1} + 1[Y_i = e], 2)]$
Final Output: $1[A_T = 1]$

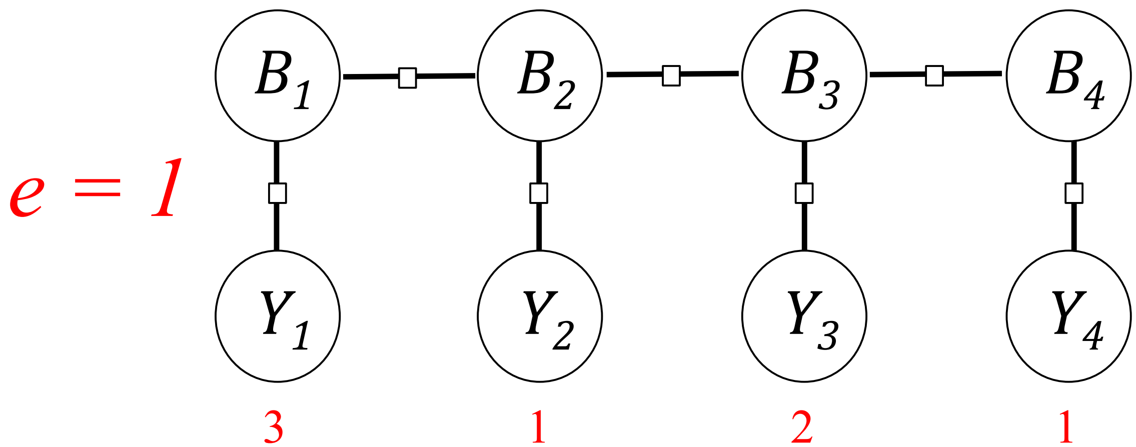
Still have factors with three variables…

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

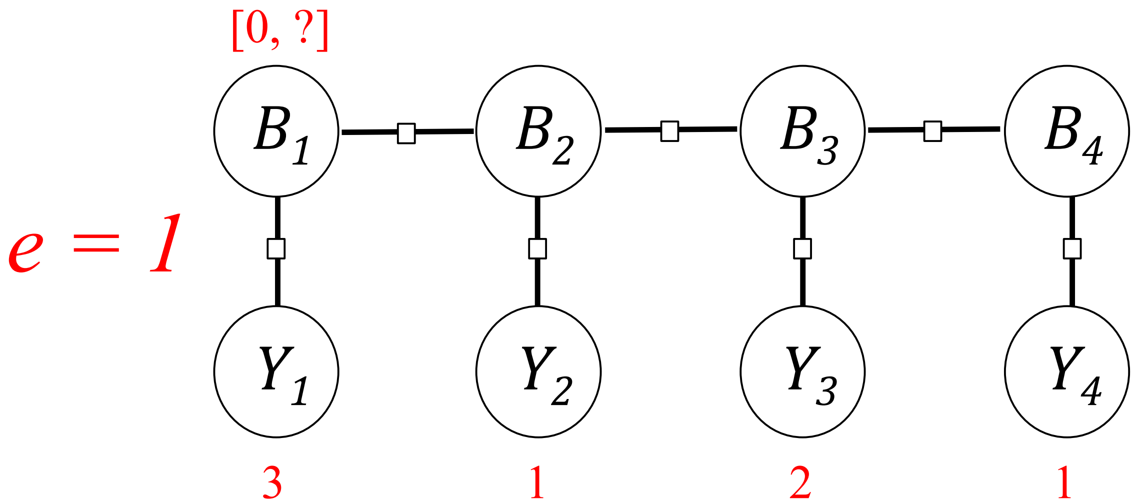


e* = *1

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

N-ary Constraints

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



Factors:

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

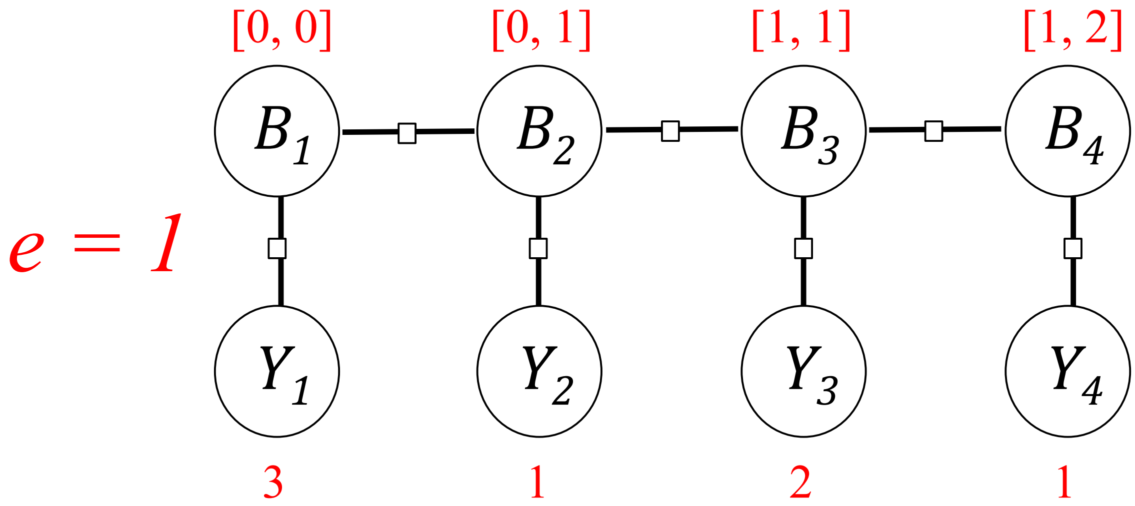
Processing: $[B_i[1] = \min(B_i[0] + 1[Y_i = e], 2)]$

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

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

N-ary Constraints

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



Factors:

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

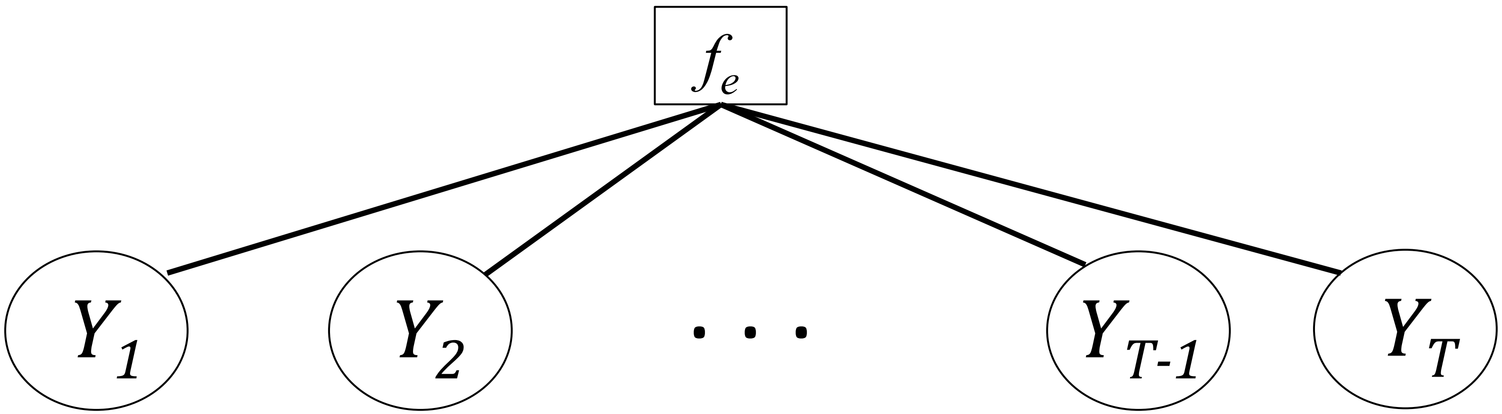
Processing: $[B_i[1] = \min(B_i[0] + 1[Y_i = e], 2)]$

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

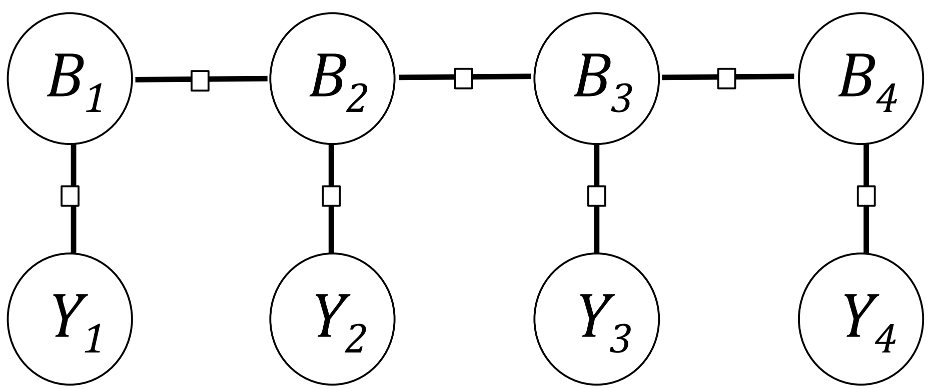
Consistency: $[B_{i-1}[1] = B_i[0]]$

N-ary Constraints

$[Y_t = e \text{ for exactly one } t]$



44



Agenda

- CSP Problem Modeling
- N-ary Constraints
- Exam Problem Solving