Inteligencia Artificial
Problemas de Satisfacción de Restricciones

Nicolás Rojas-Morales
nicolas.rojasm@usm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

Introducción

Introducción

Introducción

Optimización

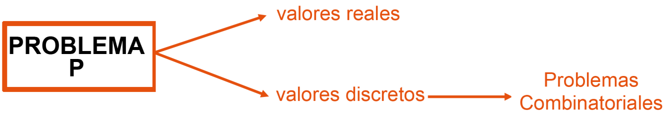
$\boxed{\textbf{PROBLEMA P}}$

Optimización

PROBLEMA P

- valores reales
- valores discretos

Optimización

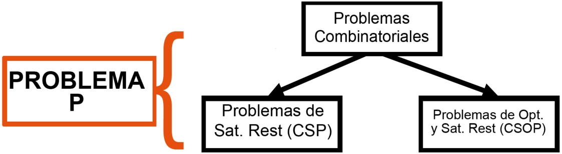


Optimización

PROBLEMA P

Un Problema Combinatorial (P) se define por:
- un conjunto de Variables $X = \{x_1,...,x_n\}$
- dominios de las variables $D_1, \dots, D_n$
- restricciones entre las variables (R)

Optimización



Optimización

PROBLEMA P

Un Problema Combinatorial (P) se define por:
- un conjunto de Variables $X = \{x_1,\ldots,x_n\}$
- dominios de las variables $D_1, \ldots, D_n$
- restricciones entre las variables (R)

$\}$ CSP

Optimización

PROBLEMA P

Un Problema Combinatorial (P) se define por:
- un conjunto de Variables $X = \{x_1,\ldots,x_n\}$
- dominios de las variables $D_1, \ldots, D_n$
- restricciones entre las variables (R)
- una función objetivo ***f*** a maximizar o minimizar

CSP

CSOP

Componentes CSP

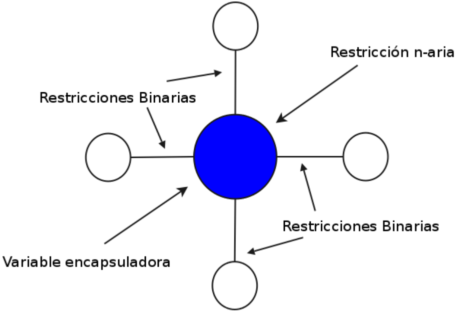
- **Variables**: Decisiones que se pueden tomar para afectar el valor de la función objetivo
- **Dominios**: Valores posibles de las variables
- **Restricciones**: Relaciones (ecuaciones e inecuaciones) que las variables están obligadas a cumplir
- **Constantes/Parámetros**: Atributos del problema conocidos a priori y fijos que permiten simplificar la formulación del modelo

- **Resolver:** Encontrar valor de las variables que satisface todas las restricciones o detectar que el problema no tiene solución

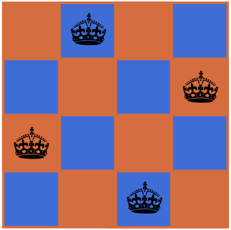
Representación de un problema

- Problema binario: Variables y restricciones forman un grafo
- Problema n-ario: Variables y restricciones forman un hiper-grafo

Método de la Variable Encapsuladora



Modelado - N Reinas



Problema: N reinas

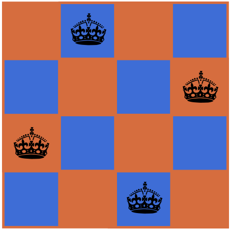
Consiste en situar N reinas en un tablero de ajedrez de NxN,
sin que se ataquen entre ellas.
Una reina ataca a otra si están en la misma fila, columna o diagonal.

Inteligencia Artificial
Problemas de Satisfacción de Restricciones

Nicolás Rojas-Morales
nicolas.rojasm@usm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

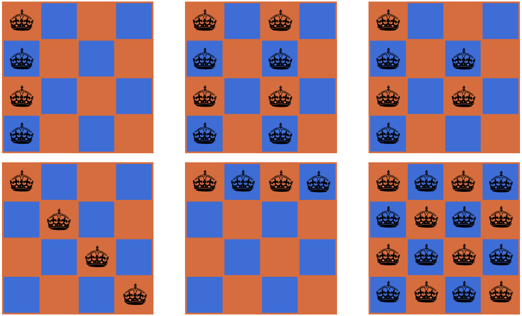
Modelado - N Reinas



Problema: N reinas

Consiste en situar N reinas en un tablero de ajedrez de NxN,
sin que se ataquen entre ellas.
Una reina ataca a otra si están en la misma fila, columna o diagonal.

Modelado - N Reinas

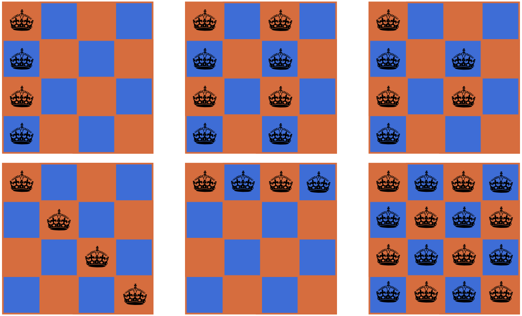


Ejemplo: N reinas

- X_{ij} - Binario
  - 1 : hay una reina en la casilla ij
  - 0 : caso contrario

$\forall i, j \in 1 \ldots 4$, si $N = 4$.

Modelado - N Reinas

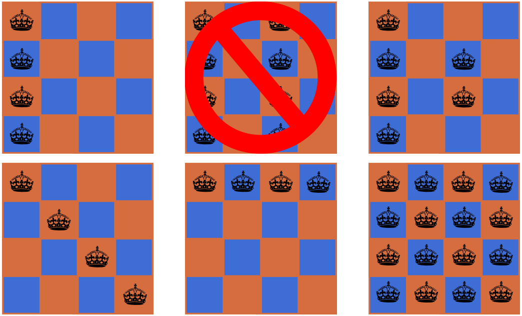


Ejemplo: N reinas

- X_k: casilla donde se encuentra la reina k

con $k \in [1, 4]$ y $X_k \in [1, 16]$

Modelado - N Reinas

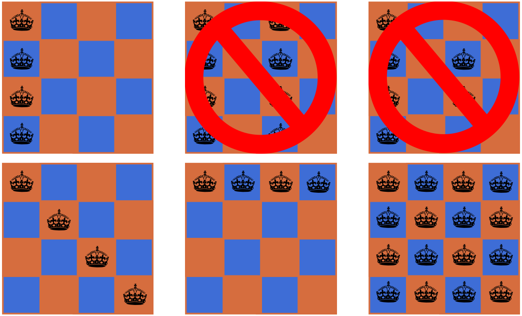


Ejemplo: N reinas

- X_k: casilla donde se encuentra la reina k

con $k \in [1, 4]$ y $X_k \in [1, 16]$

Modelado - N Reinas

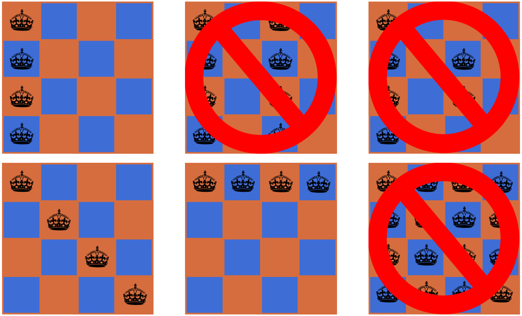


Ejemplo: N reinas

- X_k: casilla donde se encuentra la reina k

con $k \in [1, 4]$ y $X_k \in [1, 16]$

Modelado - N Reinas

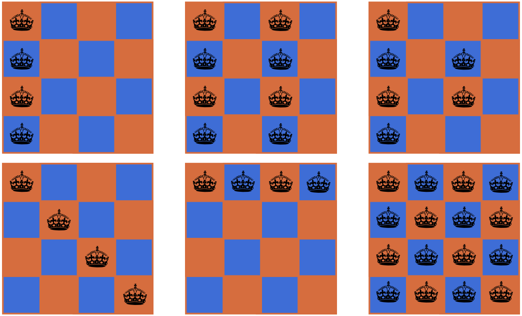


Ejemplo: N reinas

- X_k: casilla donde se encuentra la reina k

con $k \in [1, 4]$ y $X_k \in [1, 16]$

Modelado - N Reinas

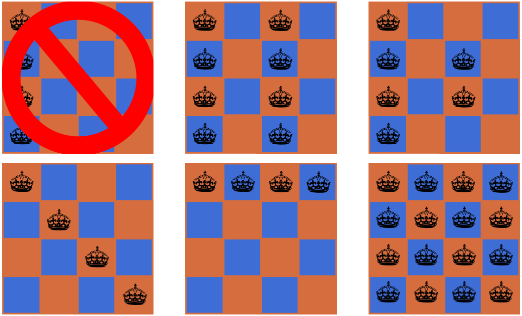


Ejemplo: N reinas

- X_i: fila donde se encuentra la reina de la columna i

con $i \in [1, 4]$ y $X_i \in [1, 4]$

Modelado - N Reinas

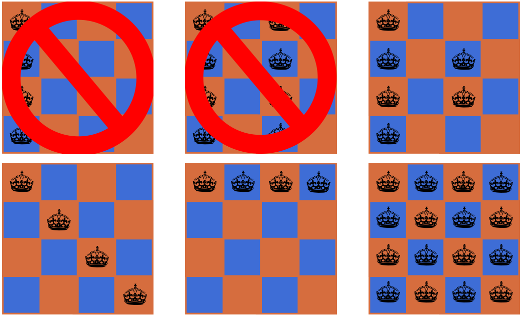


Ejemplo: N reinas

- X_i: fila donde se encuentra la reina de la columna i

con $i \in [1, 4]$ y $X_i \in [1, 4]$

Modelado - N Reinas

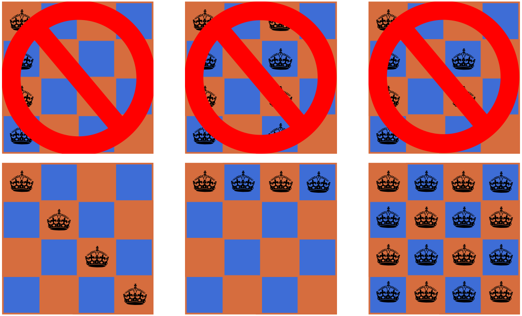


Ejemplo: N reinas

- X_i: fila donde se encuentra la reina de la columna i

con $i \in [1, 4]$ y $X_i \in [1, 4]$

Modelado - N Reinas

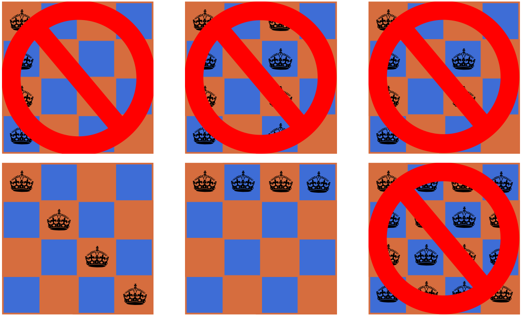


Ejemplo: N reinas

- X_i: fila donde se encuentra la reina de la columna i

con $i \in [1, 4]$ y $X_i \in [1, 4]$

Modelado - N Reinas

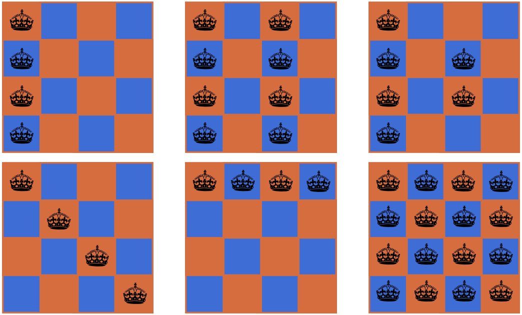


Ejemplo: N reinas

- X_i: fila donde se encuentra la reina de la columna i

con $i \in [1, 4]$ y $X_i \in [1, 4]$

Modelado - N Reinas

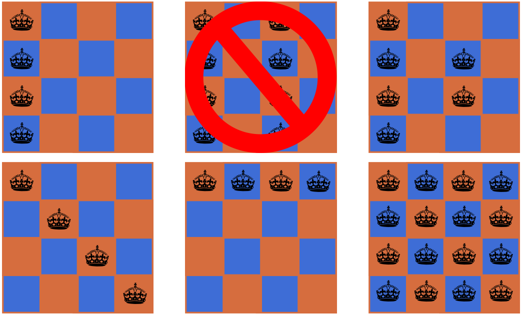


Ejemplo: N reinas

- X_j: columna donde se encuentra la reina de la fila j

con $j \in [1, 4]$ y $X_j \in [1, 4]$

Modelado - N Reinas

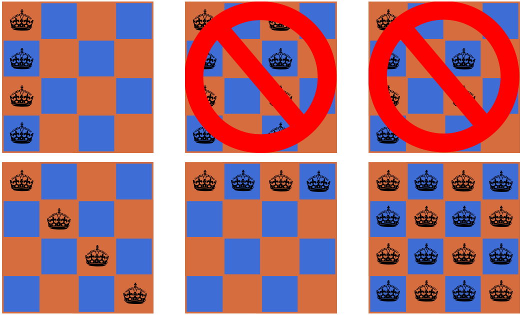


Ejemplo: N reinas

- X_j: columna donde se encuentra la reina de la fila j

con $j \in [1, 4]$ y $X_j \in [1, 4]$

Modelado - N Reinas

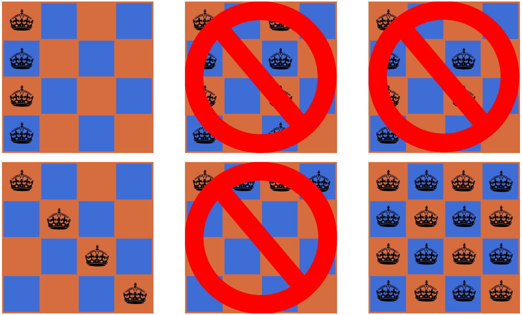


Ejemplo: N reinas

- X_j: columna donde se encuentra la reina de la fila j

con $j \in [1, 4]$ y $X_j \in [1, 4]$

Modelado - N Reinas

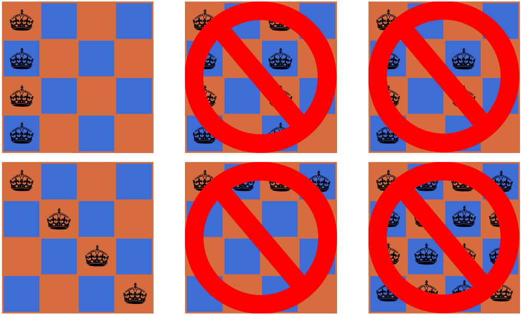


Ejemplo: N reinas

- X_j: columna donde se encuentra la reina de la fila j

con $j \in [1, 4]$ y $X_j \in [1, 4]$

Modelado - N Reinas

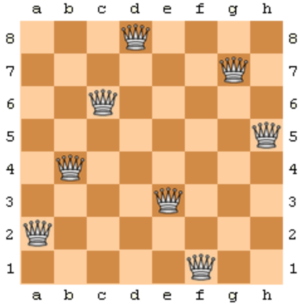


Ejemplo: N reinas

- X_j: columna donde se encuentra la reina de la fila j

con $j \in [1, 4]$ y $X_j \in [1, 4]$

Posible solución



Sudoku - Formulación

- Problema: El objetivo es rellenar una cuadrícula de n^2xn^2 celdas dividida en secciones de nxn, con valores entre 1 a n^2, considerando algunos números ya dispuestos en algunas de las celdas (pistas). No se debe repetir ningún valor en una misma fila, columna o bloque. Un sudoku está bien planteado si la solución es única.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Inteligencia Artificial
Problemas de Satisfacción de Restricciones

Nicolás Rojas-Morales
nicolas.rojasm@usm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

Sudoku - Formulación

- Problema: El objetivo es rellenar una cuadrícula de $n^2 x n^2$ celdas dividida en secciones de nxn, con valores entre 1 a n^2, considerando algunos números ya dispuestos en algunas de las celdas (pistas). No se debe repetir ningún valor en una misma fila, columna o bloque. Un sudoku está bien planteado si la solución es única.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Fase de transición

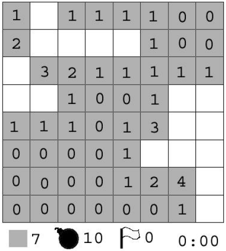
2			7					1
		1						
				5				6
6						4		
7								
			6					
	5							2

2		5	7	4			6	1
4		1			5			7
8	7		1	9	2	5	4	
		7		5			1	6
6		2	3		7	4		5
				1	6		3	
7		3	5					9
9		8	6		1	3		
	5			3		6		2

2		5	7	4	3	8	6	1
4	3	1	8		5	9	2	7
8	7	6	1	9	2	5	4	
	8	7		5	9		1	6
6	1	2	3	8	7	4	9	5
5	4	9	2	1	6	7	3	8
7	6	3	5		4		8	9
9		8	6	7	1	3	5	4
1	5	4	9	3	8	6		2

Buscaminas - Formulación

- Problema: Identificar las casillas que contienen una bomba a partir de la información desplegada por sus casillas adyacentes.



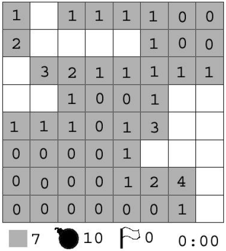
Inteligencia Artificial
Problemas de Satisfacción de Restricciones

Nicolás Rojas-Morales
nicolas.rojasm@usm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

Buscaminas - Formulación

- Problema: Identificar las casillas que contienen una bomba a partir de la información desplegada por sus casillas adyacentes.



Car Sequencing - Formulación

- Secuencia de vehículos en una línea de ensamblaje
- Cada vehículo tiene una lista de opciones a ser instaladas (Radio, Techo plegable, Barras para llevar bicicletas, etc).
- Cada opción es instalada por una estación diferente cuya capacidad de operación es limitada
- Por ejemplo, por cada q automóviles, sólo p_{barra} vehículos pueden requerir barras
- Cada combinación de opciones constituye una clase de vehículo
- Existe una demanda de vehículos de cada clase
- El objetivo es encontrar un orden en la secuencia de vehículos sin exceder la capacidad de cada estación y cumplir con la demanda

El problema de la Cebra

En una calle, las cinco primeras casas son de diferentes colores, en dichas casas viven personas de diferentes nacionalidades, con diferentes mascotas, gustan de diferentes bebidas y practican diferentes deportes.
Además se consideran las siguientes restricciones:

1. El inglés vive en la casa roja
2. El español tiene un perro
3. El hombre de la casa verde bebe café
4. El alemán bebe té
5. La casa verde está a la derecha de la casa celeste
6. El jugador de tenis tiene una granja de hormigas
7. El hombre de la casa amarilla juega volleyball
8. El hombre de la casa del medio bebe leche
9. El nigeriano vive en la primera casa
10. El jugador de fútbol vive cerca del hombre que tiene un lobo
11. El jugador de volleyball vive cerca del dueño del gato
12. El basquetbolista bebe jugo de naranja
13. El japonés practica polo
14. El nigeriano vive cerca de la casa lila

La pregunta es:
¿Quién es el dueño de la cebra? y ¿Quién bebe cervezas?

Inteligencia Artificial
Problemas de Satisfacción de Restricciones

Nicolás Rojas-Morales
nicolas.rojasm@usm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

El problema de la Cebra

En una calle, las cinco primeras casas son de diferentes colores, en dichas casas viven personas de diferentes nacionalidades, con diferentes mascotas, gustan de diferentes bebidas y practican diferentes deportes.
Además se consideran las siguientes restricciones:

1. El inglés vive en la casa roja
2. El español tiene un perro
3. El hombre de la casa verde bebe café
4. El alemán bebe té
5. La casa verde está a la derecha de la casa celeste
6. El jugador de tenis tiene una granja de hormigas
7. El hombre de la casa amarilla juega volleyball
8. El hombre de la casa del medio bebe leche
9. El nigeriano vive en la primera casa
10. El jugador de fútbol vive cerca del hombre que tiene un lobo
11. El jugador de volleyball vive cerca del dueño del gato
12. El basquetbolista bebe jugo de naranja
13. El japonés practica polo
14. El nigeriano vive cerca de la casa lila

La pregunta es:
¿Quién es el dueño de la cebra? y ¿Quién bebe cervezas?

Análisis

Conjunto de atributos:

- Nacionalidades = { Inglés, Español, Irlandés, Nigeriano, Japonés}

N_1	N_2	N_3	N_4	N_5
Inglés	Español	Irlandés	Nigeriano	Japonés

- Colores = { Roja, Verde, Celeste, Amarilla, Azul}

C_1	C_2	C_3	C_4	C_5
Roja	Verde	Celeste	Amarilla	Azul

- Deportes = { Tenis, Voleyball, Futbol, Basketball, Polo}

D_1	D_2	D_3	D_4	D_5
Tenis	Voleyball	Futbol	Basketball	Polo

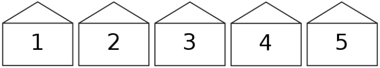
- Bebidas = { Café, Té, Leche, Jugo de Naranja, Cervezas}

B_1	B_2	B_3	B_4	B_5
Café	Té	Leche	Jugo de Naranja	Cervezas

- Mascotas = { Perro, Caracoles, Lobo, Caballo, Cebra}

M_1	M_2	M_3	M_4	M_5
Perro	Caracoles	Lobo	Caballo	Cebra

Planteo I



Considere:

- La persona que vive en la primera casa es 1
- ...
- La persona que vive en la última casa es 5

Ejemplo de planteo de restricciones:

- El Inglés vive en la casa Roja
  Es decir, la persona de nacionalidad inglesa (N_1) es la misma que vive en la casa roja (C_1)
  $$N_1 = C_1$$
- La casa Verde está a la derecha de la casa Celeste
  Es decir, la persona de la casa Verde (C_2) debe ser mayor que la persona de la casa Celeste (C_3)
  $$C_2 > C_3$$

Planteo II

- El hombre de la casa del medio bebe Leche
  Es decir, la persona que bebe Leche (B_3) es 3

$$B_3 = 3$$

- El jugador de Fútbol vive cerca del hombre que tiene un Lobo
  Importante: ¿Qué es vivir cerca?
  $\rightarrow$ Vivir cerca significa que viven a lo más a una casa de distancia

$$|D_3 - M_3| = 1$$

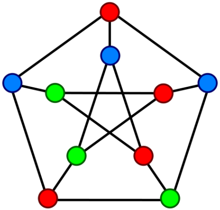
CSOP: Constraint Satisfaction Optimization Problems

CSOP

Un CSOP consiste en un CSP estandar P y una función objetivo f definida sobre el conjunto de variables X que mapea una solución de P a un número real. La solución a un CSOP es una solución de P que optimiza el valor de la función objetivo.

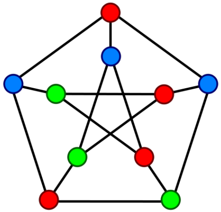
CSP vs CSOP: Coloreo de Grafos

- Asignar distintos colores a los vértices de un grafo de manera que ningún par de vértices adyacentes compartan el color.



CSP vs CSOP: Coloreo de Grafos

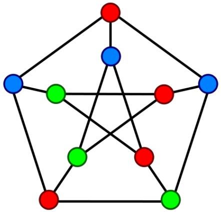
- Asignar distintos colores a los vértices de un grafo de manera que ningún par de vértices adyacentes compartan el color.



- CSP
  - Considerando un conjunto de colores dado

CSP vs CSOP: Coloreo de Grafos

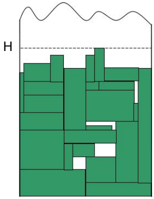
- Asignar distintos colores a los vértices de un grafo de manera que ningún par de vértices adyacentes compartan el color.



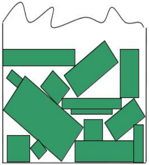
- CSP
  - Considerando un conjunto de colores dado
- CSOP
  - Minimizando la cantidad de colores

CSP vs CSOP: 2D Space Planning

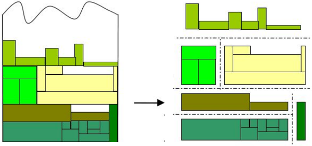
- Un conjunto de rectángulos deben ser ubicados en un contenedor rectangular de mayor tamaño.



Ortogonal



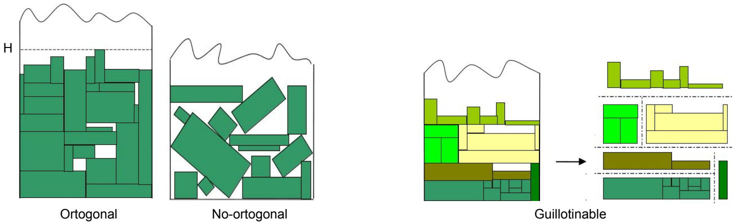
No-ortogonal



Guillotinable

CSP vs CSOP: 2D Space Planning

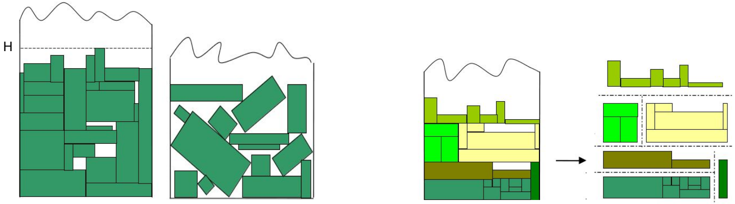
- Un conjunto de rectángulos deben ser ubicados en un contenedor rectangular de mayor tamaño.



- CSP
  - El contenedor tiene un ancho W y altura H
  - El objetivo es ubicar todos los rectángulos en el contenedor sin que se sobrepongan.

CSP vs CSOP: 2D Space Planning

- Un conjunto de rectángulos deben ser ubicados en un contenedor rectangular de mayor tamaño.



Ortogonal   No-ortogonal   Guillotinable

- CSP
  - El contenedor tiene un ancho W y altura H
  - El objetivo es ubicar todos los rectángulos en el contenedor sin que se sobrepongan.
- CSOP
  - El contenedor tiene un ancho W y altura infinita
  - El objetivo es minimizar la altura del contenedor luego de colocar todos los rectángulos.

University Course Timetabling Problem (1/2)

Bloque	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
1 2						
3 4		INF236			INF295	
5 6	FIS120		FIS120			
7 8			INF266			
9 10		INF236			FIS120	
11 12						
13 14	INF246		INF246			
15 16						
17 18						

University Course Timetabling Problem (2/2)

Dado:

- Un conjunto de eventos
- Un conjunto de timeslots
- Un conjunto de salas
- Un conjunto de características
- Un conjunto de estudiantes que asisten a ciertos eventos en ciertas salas en ciertos timeslots

Se requiere:

- Cada evento debe ser asignado a una sala que posea las características y capacidad requerida
- Cada sala debe ser utilizada por a lo más un evento en cada timeslot
- Eventos con alumnos en común deben ser asignados a diferentes timeslots

Se desea:

- Los estudiantes no tengan eventos en el último timeslot de cada día
- Los estudiantes no tengan más de dos timeslots seguidos de eventos
- Los estudiantes no tengan un único evento en un día.

Glosario

- CSP-CSOP
- Restricción Unaria
- Restricción Binaria
- Restricción N-aria
- CSP Binario
- CSP N-ario
- Grafo de Restricciones
- Variable Encapsuladora
- Fase de Transición
- Restricciones blandas y duras