

Investigación de Operaciones [INF-3144]

Capítulo 2: Programación con Restricciones

Dr. Ricardo Soto

[ricardo.soto@ucv.cl]

[<http://www.inf.ucv.cl/~rsoto>]

Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

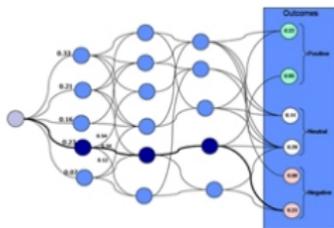


PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO



1. Introducción

Es una **tecnología** que tiene sus raíces en diversas áreas...



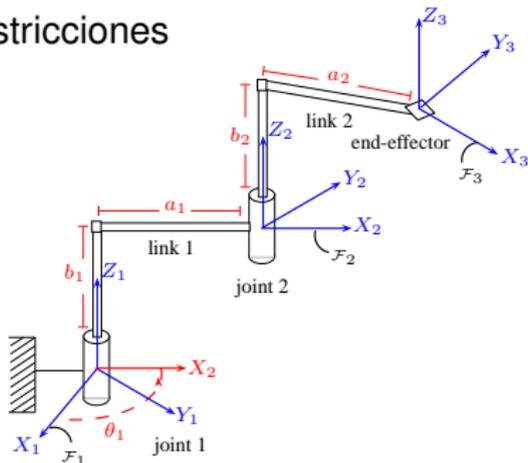
```
1 package main
2 import (
3     "fmt"
4     "math/rand"
5     "time"
6 )
7
8 public class Main {
9     public static void main(String[] args) {
10         // Crear un objeto de tipo Main
11         Main main = new Main();
12         // Ejecutar el método run
13         main.run();
14     }
15
16     // Método run
17     public void run() {
18         // Crear un objeto de tipo Random
19         Random random = new Random();
20         // Crear un objeto de tipo Time
21         Time time = new Time();
22         // Crear un objeto de tipo Main
23         Main main = new Main();
24         // Ejecutar el método run
25         main.run();
26     }
27
28     // Método run
29     public void run() {
30         // Crear un objeto de tipo Random
31         Random random = new Random();
32         // Crear un objeto de tipo Time
33         Time time = new Time();
34         // Crear un objeto de tipo Main
35         Main main = new Main();
36         // Ejecutar el método run
37         main.run();
38     }
39 }
40
41 // Fin del código
```

Objetivo?

Resolver problemas que se puedan representar en función de **variables** y **restricciones**

Ejemplos Reales

- Detección de errores de precisión en robots (IRCCYN Lab)
...+ de 500 variables y restricciones
- Diseño de un sistema de aire acondicionado para aviones (Dassault Aviation)
...+ de 1000 variables y restricciones



3. Proyectos Resueltos por alumnos PUCV



- Manufacturing Cell Design
- Juan Gutiérrez, Alexis López

3. Proyectos Resueltos por alumnos PUCV



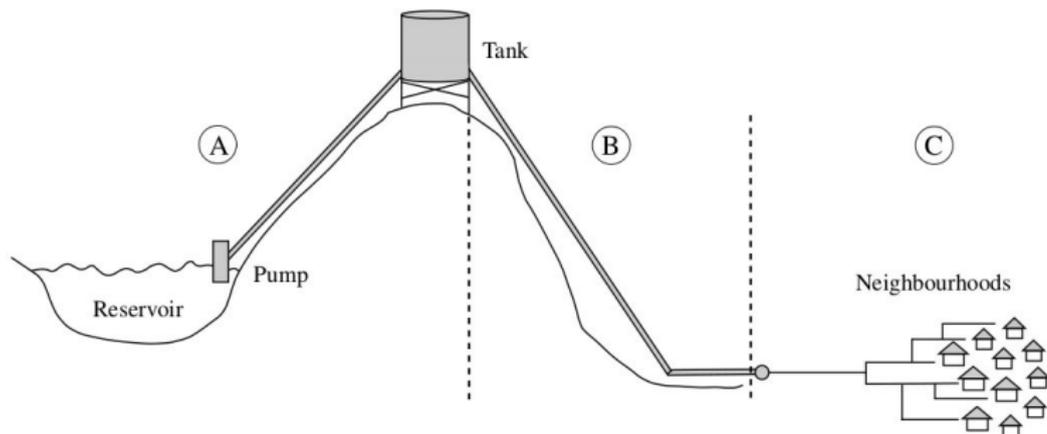
- Nurse Rostering
- Renzo Pizarro, Gianni Rivera

3. Proyectos Resueltos por alumnos PUCV



- Ms Pacman Problem
- Francisco Lobos, Diego González

3. Proyectos Resueltos por alumnos PUCV



- Water Distribution
- Paz Clayton, Ricardo Rojas

3. Proyectos Resueltos por alumnos PUCV



- Portfolio Selection
- Camila Allendes, Hans Berendsen

3. Proyectos Resueltos por alumnos PUCV



- Open-pit mining
- Boris Almonacid

4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

Ejemplo 1

Resolver la siguiente ecuación, reemplazando las letras por dígitos distintos.

$$\begin{array}{rcccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

Ejemplo 1

Resolver la siguiente ecuación, reemplazando las letras por dígitos distintos.

$$\begin{array}{rcccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

$$\begin{array}{rcccc} & 9 & 5 & 6 & 7 \\ + & 1 & 0 & 8 & 5 \\ \hline 1 & 0 & 6 & 5 & 2 \end{array}$$

4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

Modelo

- Variables

$S, E, N, D, M, O, R, Y \in [0, 9]$

- Restricciones

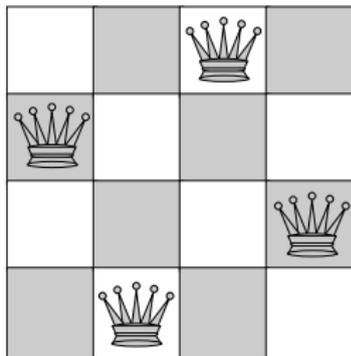
$$\begin{array}{rcccccccc} & & & 1000 \cdot S & + & 100 \cdot E & + & 10 \cdot N & + & D \\ & & & + & 1000 \cdot M & + & 100 \cdot O & + & 10 \cdot R & + & E \\ = & 10000 \cdot M & + & 1000 \cdot O & + & 100 \cdot N & + & 10 \cdot E & + & Y \end{array}$$

$$S \neq E, S \neq N, S \neq D \dots R \neq Y$$

4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

Ejemplo 2 - N-Queens

Ubicar n reinas en un tablero de ajedrez de $n \times n$, de manera tal que no se puedan atacar.



4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

Modelo

- Variables

$$Q_1, Q_2, Q_3, Q_4 \in [1, 4]$$

- Restricciones (para $i \in [1, 3]$ y $j \in [i + 1, 4]$)

$$Q_i \neq Q_j \text{ (filas)}$$

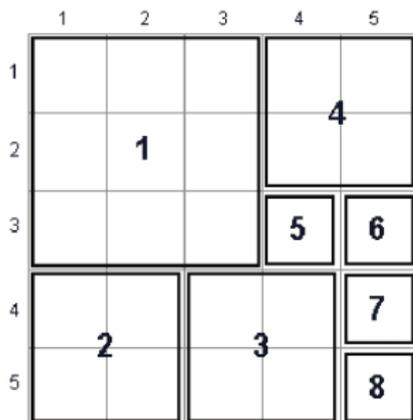
$$Q_i + i \neq Q_j + j \text{ (diagonal 1)}$$

$$Q_i - i \neq Q_j - j \text{ (diagonal 2)}$$

4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

Ejercicio 1 - Packing Squares

Ubicar un conjunto de cuadrados dentro una base cuadrada de tal manera que ningún cuadrado se traslapa con otro.



4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

- Variables

$$x_1, x_2, \dots, x_{squares} \in [1, sideSize]$$

$$y_1, y_2, \dots, y_{squares} \in [1, sideSize]$$

- Constantes

sideSize

squares

size₁, size₂, ..., size_{squares}

- Restricciones (para $i \in [1, squares]$) //inside

$$x_i \leq sideSize - size_i + 1$$

$$y_i \leq sideSize - size_i + 1$$

4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

- Restricciones (para $i \in [1, \text{squares}]$ y $j \in [i + 1, \text{squares}]$)
//noOverlap

$$x_i + \text{size}_i \leq x_j \text{ OR}$$

$$x_j + \text{size}_j \leq x_i \text{ OR}$$

$$y_i + \text{size}_i \leq y_j \text{ OR}$$

$$y_j + \text{size}_j \leq y_i$$

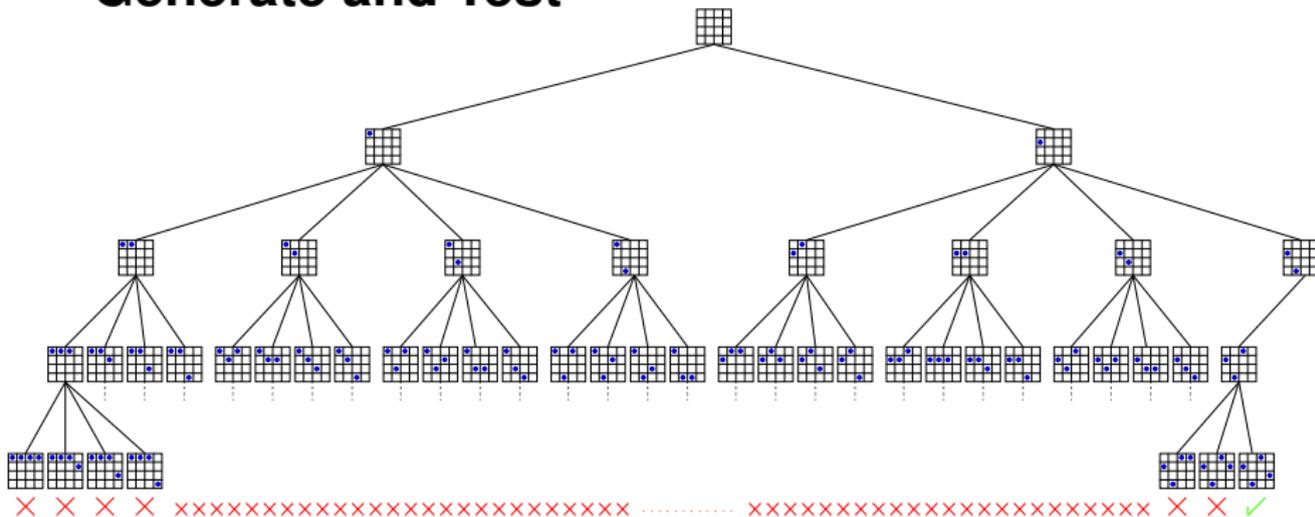
4. Problema de Satisfacción de Restricciones (Constraint Satisfaction Problem, CSP)

A Constraint Satisfaction Problem \mathcal{P} is defined by a triple $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$ where:

- \mathcal{X} is a n -tuple of variables $\mathcal{X} = \langle x_1, x_2, \dots, x_n \rangle$,
- \mathcal{D} is a corresponding n -tuple of domains $\mathcal{D} = \langle D_1, D_2, \dots, D_n \rangle$ such that $x_i \in D_i$, and D_i is a set of values, for $i = 1, \dots, n$.
- \mathcal{C} is a m -tuple of constraints $\mathcal{C} = \langle C_1, C_2, \dots, C_m \rangle$.

Solving = Modeling + Search

Generate and Test



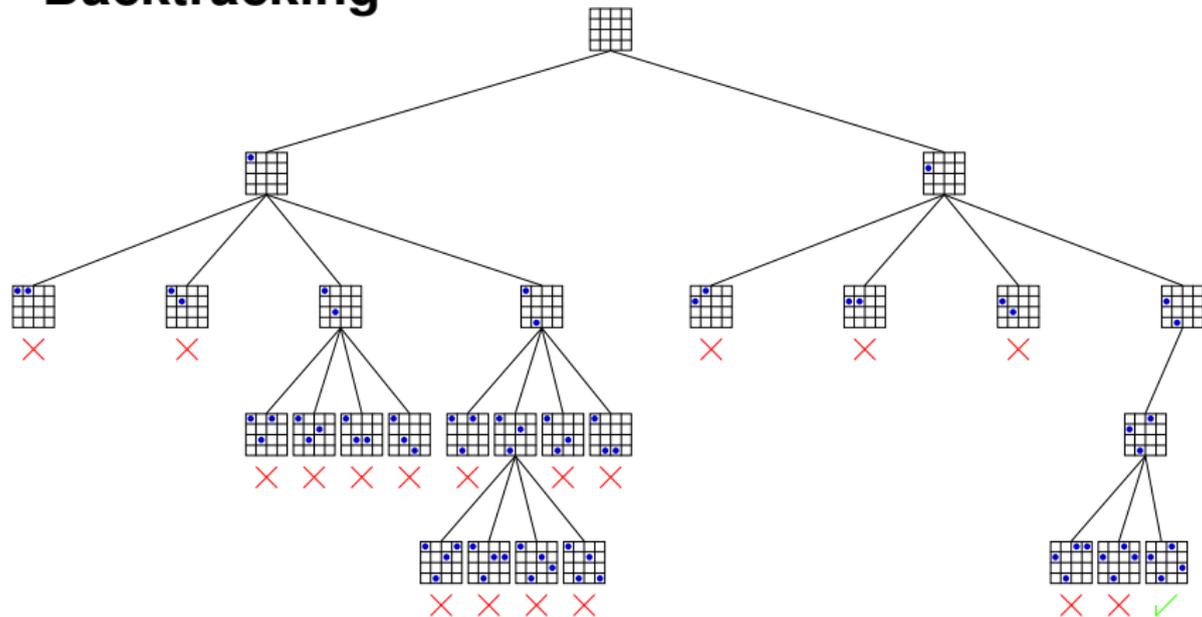
Problemas

- Gran cantidad de instanciaciones que no conducen a una solución
- Las restricciones se evalúan con todas las variables instanciadas

Solución?

- Evaluar las restricciones apenas se instancien las variables involucradas.

Backtracking



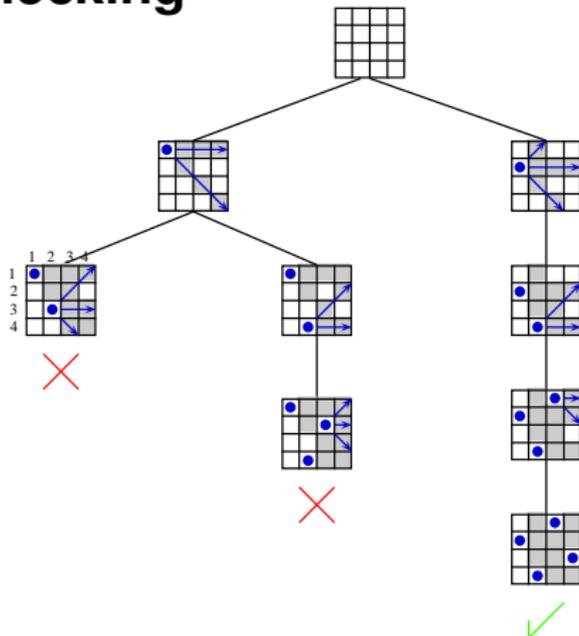
Principal Problema

- No se pueden detectar inconsistencias sin instanciar todas las variables involucradas en una restricción.

Solución?

- Eliminar valores temporalmente de los dominios utilizando técnicas de consistencia (arc-consistency).

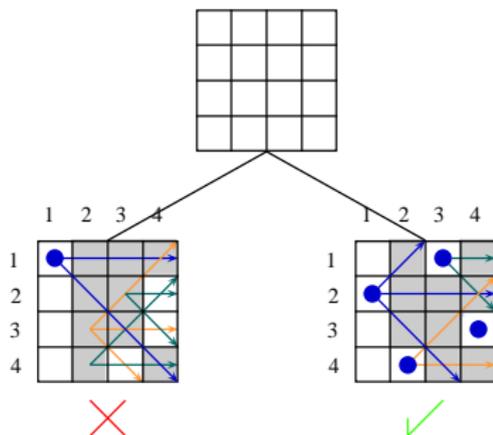
Forward Checking



Se puede mejorar?

- Verificar no sólo la consistencia entre la variable actual y las futuras, sino que también entre las futuras...

Maintaining Arc Consistency (Full Look Ahead)

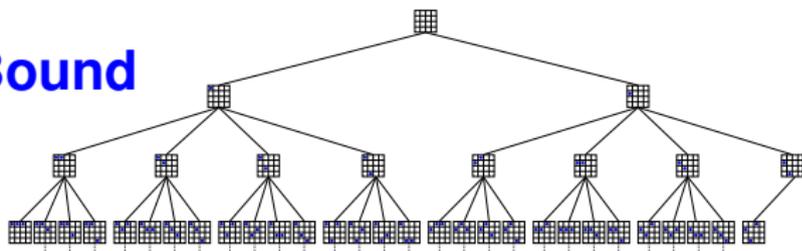


Optimización

- Basta con extender el algoritmo de búsqueda para considerar la función objetivo

Algoritmo más utilizado para optimización en CP:

Branch and Bound



6. Heurísticas de selección de variable y valor

Variable

- First-fail (dominio más pequeño)
- Most-constrained variable
- Reduce-first (dominio más grande)
- Round-robin (orden equitativo, por ej. de la 1era a la última)

Valor

- smallest
- median
- maximal

Diversos Lenguajes para CP

- Basados en programación lógica (**Eclipse**, **SicstusProlog...**)
- Basados en programación orientada a objetos (**ILOG**, **Gecode...**)
- Modelado de alto nivel (**OPL**, **Zinc...**)