

# Uma Introdução à Programação por Restrições

Pedro Henrique Del Bianco Hokama

1 Programação por Restrições

2 Como funciona?

3 Histórico

4 Exemplos

1 / 56

2 / 56

## O que é?

É um paradigma de programação que define variáveis e relações entre essas variáveis através de restrições.

O usuário define quais as características que uma solução deve satisfazer.

“O que resolver” vs “Como resolver”

3 / 56

## Problema de Satisfação de Restrições

No CSP (Constraint Satisfaction Problem) temos:

- Um conjunto de variáveis  $X = \{x_1, \dots, x_n\}$ ;
- Um conjunto finito  $D_i$  para cada variável  $x_i$ , com todos os valores que podem ser atribuídos a variável  $x_i$ ;
- Um conjunto de restrições entre as variáveis que devem ser satisfeitas.

4 / 56

## Problema de Satisfação de Restrições

Podemos estar interessados em:

- Encontrar qualquer solução;
- Encontrar todas as soluções;
- Provar que não existe nenhuma solução;
- Dado uma função objetivo, encontrar uma solução boa ou ótima.

## Sudoku

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

$x_{i,j} \in \{1, 2, \dots, 9\}$

5 / 56

6 / 56

## Programação por Restrições

- Inteligência Artificial
- Visão Computacional
- Pesquisa Operacional
- Sistemas Especialistas
- etc..

7 / 56

8 / 56

## Programação por Restrições

Como utilizar programação por restrições?

- Descrever o Problema
- Modelar o Problema
  - ▶ Variáveis
  - ▶ Domínios
  - ▶ Restrições
- Resolver o Problema

Exemplo:

9 / 56

## Modelar

Variáveis:

- $x$  e  $y$

Domínios:

- Domínio de  $x$  é  $[5 \dots 12]$
- Domínio de  $y$  é  $[2 \dots 17]$

Restrições:

- $x + y = 17$
- $x - y = 5$

Quais os valores de  $x$  e  $y$ , onde  $x$  é um inteiro entre 5 e 12, e  $y$  é um inteiro entre 2 e 17 tal que  $x + y = 17$  e  $x - y = 5$ .

11 / 56

## Descrever o problema

Quais os valores de  $x$  e  $y$ , onde  $x$  é um inteiro entre 5 e 12, e  $y$  é um inteiro entre 2 e 17 tal que  $x + y = 17$  e  $x - y = 5$ .

10 / 56

## Resolver

- Explorar o espaço de busca para encontrar uma solução. (Todas as combinações de valores)

Técnicas para encontrar uma solução:

- Busca construtiva (Backtracking)
- Propagação
  - ▶ Propagação inicial
  - ▶ Propagação durante a busca

12 / 56

## Propagação Inicial

Domínio antes da propagação:

$$D(x) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

$$D(y) = [2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17]$$

- $x + y = 17$

13 / 56

## Propagação Inicial

Domínio:

$$D(x) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

$$D(y) = [2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ ~~13 \ 14 \ 15 \ 16 \ 17~~]$$

- $x + y = 17$   
 $y = 17 - x, x \geq 5$   
portanto  $y \leq 17 - 5$   
 $y \leq 12$

15 / 56

## Propagação Inicial

Domínio:

$$D(x) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

$$D(y) = [2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17]$$

- $x + y = 17$   
 $y = 17 - x, x \geq 5$   
portanto  $y \leq 17 - 5$   
 $y \leq 12$

14 / 56

## Propagação Inicial

Domínio:

$$D(x) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

$$D(y) = [2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

- $x + y = 17$   
 $y = 17 - x, x \leq 12$   
portanto  $y \geq 17 - 12$   
 $y \geq 5$

16 / 56

## Propagação Inicial

Domínio:

$$D(x) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

$$D(y) = [\del{2} \ \del{3} \ \del{4} \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

- $x + y = 17$   
 $y = 17 - x, x \leq 12$   
portanto  $y \geq 17 - 12$   
 $y \geq 5$

17 / 56

## Propagação Inicial

Domínio:

$$D(x) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

$$D(y) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

- $x - y = 5$   
 $x = 5 + y, y \geq 5$   
portanto  $x \geq 5 + 5$   
 $x \geq 10$

18 / 56

## Propagação Inicial

Domínio:

$$D(x) = [\del{5} \ \del{6} \ \del{7} \ \del{8} \ \del{9} \ 10 \ 11 \ 12]$$

$$D(y) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

- $x - y = 5$   
 $x = 5 + y, y \geq 5$   
portanto  $x \geq 5 + 5$   
 $x \geq 10$

19 / 56

## Propagação Inicial

Domínio:

$$D(x) = [10 \ 11 \ 12]$$

$$D(y) = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12]$$

- $x + y = 17$   
 $y = 17 - x, x \geq 10$   
portanto  $y \leq 17 - 10$   
 $y \leq 7$

20 / 56

## Propagação Inicial

Domínio:

$$D(x) = [10 \ 11 \ 12]$$

$$D(y) = [5 \ 6 \ 7 \ ~~8 \ 9 \ 10 \ 11 \ 12~~]$$

- $x + y = 17$   
 $y = 17 - x, x \geq 10$   
portanto  $y \leq 17 - 10$   
 $y \leq 7$

21 / 56

## Propagação Inicial

Domínio:

$$D(x) = [10 \ 11 \ 12]$$

$$D(y) = [5 \ 6 \ 7]$$

22 / 56

## Busca Construtiva

Domínio:

$$D(x) = [10 \ 11 \ 12]$$

$$D(y) = [5 \ 6 \ 7]$$

23 / 56

## Busca Construtiva

Atribuição:

$$x = 10$$

$$D(y) = [5 \ 6 \ 7]$$

Propagação Durante a Busca:

- $x + y = 17$

24 / 56

## Busca Construtiva

Domínio local:

$$x = 10$$

$$D(y) = [5 \del{6} 7]$$

Propagação Durante a Busca:

- $x + y = 17$

25 / 56

## Propagação Durante a Busca

Domínio local:

$$x = 10$$

$$D(y) = [7]$$

- $x - y = 5$

26 / 56

## Propagação Durante a Busca

Domínio local:

$$x = 10$$

$$D(y) = [] \text{ falha!}$$

- $x - y = 5$

27 / 56

## Backtracking

Domínio local:

$$x = 11$$

$$D(y) = [5 \ 6 \ 7]$$

28 / 56

## Propagação Durante a Busca

Domínio local:

$$x = 11$$

$$D(y) = [5 \ 6 \ 7]$$

Propagação Durante a Busca:

- $x + y = 17$

29 / 56

## Propagação Durante a Busca

Domínio local:

$$x = 11$$

$$D(y) = [\cancel{5} \ 6 \ \cancel{7}]$$

Propagação Durante a Busca:

- $x + y = 17$

30 / 56

## Propagação Durante a Busca

Domínio local:

$$x = 11$$

$$D(y) = [6]$$

Propagação Durante a Busca:

- $x - y = 5$

31 / 56

## Propagação Durante a Busca

Domínio local:

$$x = 11$$

$$D(y) = [6]$$

Propagação Durante a Busca:

- $x - y = 5$  não é violado

32 / 56



## Solução

$$x = 11$$

$$y = 6$$

Está no domínio, não viola nenhuma restrição, portanto é uma solução.

33 / 56

## Histórico

- Na grécia antiga...

34 / 56



35 / 56

## Epitáfio de Diofanto - Século II

"Deus deu à ele um sexto de sua vida na infância, Um duodécimo como adolescente enquanto cresciam bigodes; E ainda um sétimo antes de iniciar o casamento; Em cinco anos chegou um vigoroso filho. Ah! Querida criança do mestre e sábio, Depois de alcançar metade da idade que viveu seu pai, o destino frio o levou. Após consolar-se por quatro anos com a ciência dos números, ele terminou sua vida."

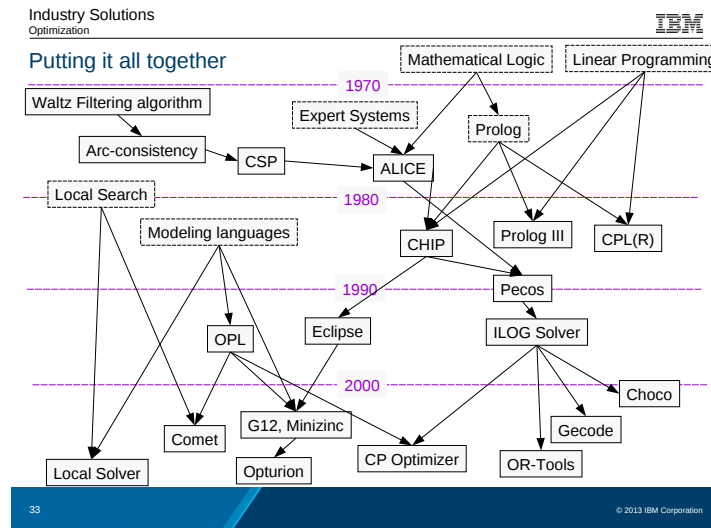
$$x = \frac{x}{6} + \frac{x}{12} + \frac{x}{7} + 5 + \frac{x}{2} + 4$$

$$x = 84$$

36 / 56

## Histórico

- 1956 - Início da Inteligência Artificial (McCarthy no Dartmouth workshop)
- 1963 - Início do CP (Sutherland em sua tese de doutorado)
- 1965 - Primeiros algoritmos de backtracking



37 / 56

## Resolvedores

- CP Optimizer - IBM
- Choco solver (biblioteca Java)
- Gecode (biblioteca C++)
- [OR-Tools - Google](#)
- ...

39 / 56

## Operadores e Restrições - OR-Tools

- Variáveis: Booleanas, Intervalo (de duração fixa ou não), Inteira.
- Operadores Matemáticos  
+ - \\* min max log abs ...
- Operadores Lógicos  
and or ==> == ...
- Especializadas  
AllDifferent, AllDifferentExcept, AllowedAssignments, AtMost, IndexOf, Element, StartAfter, StartBefore, EndsAfter, ...

40 / 56

## OR-Tool - google

Problema: definir valores para os dígitos  $c$ ,  $p$ ,  $i$ ,  $s$ ,  $f$ ,  $u$  e  $n$ , tal que essa soma seja verdadeira:

$$\begin{array}{r} c \ p \\ + \ i \ s \\ \hline f \ u \ n \end{array}$$

Todos os dígitos devem ser diferentes, e  $f \neq 0$ .

```
CpModelBuilder cp_model;

// Define decision variables.
Domain digit(0, 9);
Domain digit_sem_zero(1, 9);

IntVar c = cp_model.NewIntVar(digit).WithName("C");
IntVar p = cp_model.NewIntVar(digit).WithName("P");
IntVar i = cp_model.NewIntVar(digit).WithName("I");
IntVar s = cp_model.NewIntVar(digit).WithName("S");
IntVar f = cp_model.NewIntVar(digit_sem_zero).WithName("F");
IntVar u = cp_model.NewIntVar(digit).WithName("U");
IntVar n = cp_model.NewIntVar(digit).WithName("N");
// [END variables]

// [START constraints]
cp_model.AddAllDifferent({c, p, i, s, f, u, n});
cp_model.AddEquality(
    LinearExpr::ScalProd({c, p, i, s},
                        {10, 1, 10, 1}),
    LinearExpr::ScalProd({f, u, n},
                        {100, 10, 1}));
```

41 / 56

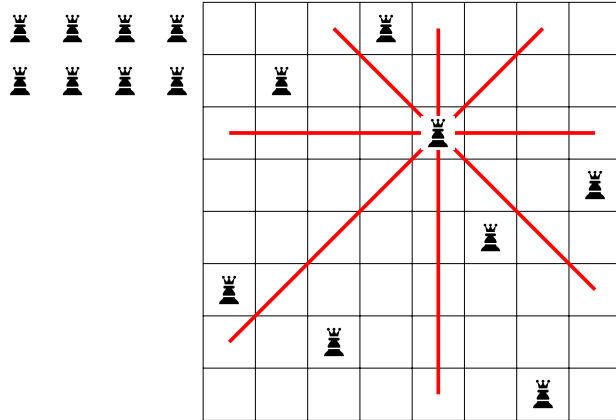
```
Model model;

const CpSolverResponse response = SolveCpModel(cp_model.Build(), &model);
std::cout << " C=" << SolutionIntegerValue(response, c) << " "
  << "P=" << SolutionIntegerValue(response, p) << " " << std::endl
  << " I=" << SolutionIntegerValue(response, i) << " "
  << "S=" << SolutionIntegerValue(response, s) << " " << std::endl
  << "F=" << SolutionIntegerValue(response, f) << " "
  << "U=" << SolutionIntegerValue(response, u) << " "
  << "N=" << SolutionIntegerValue(response, n) << std::endl;
```

```
make run SOURCE=pasta/cpisfun.cpp
```

```
C=2 P=3
I=8 S=4
F=1 U=0 N=7
```

## Problema das 8 rainhas



```

from ortools.constraint_solver import pywrapcp

solver = pywrapcp.Solver("n-queens")

n=8
q = [solver.IntVar(0, n - 1, "q%i" % i) for i in range(n)]

solver.Add(solver.AllDifferent(q))

for i in range(n):
    for j in range(i):
        solver.Add(q[i] != q[j])
        solver.Add(q[i] + i != q[j] + j)
        solver.Add(q[i] - i != q[j] - j)

db = solver.Phase([q[i] for i in range(n)],
                  solver.CHOOSE_MIN_SIZE_LOWEST_MAX,
                  solver.ASSIGN_CENTER_VALUE)

solver.NewSearch(db)

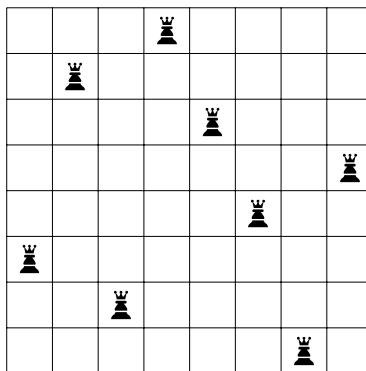
solver.NextSolution()
print([q[i] for i in range(n)])
solver.EndSearch()

```

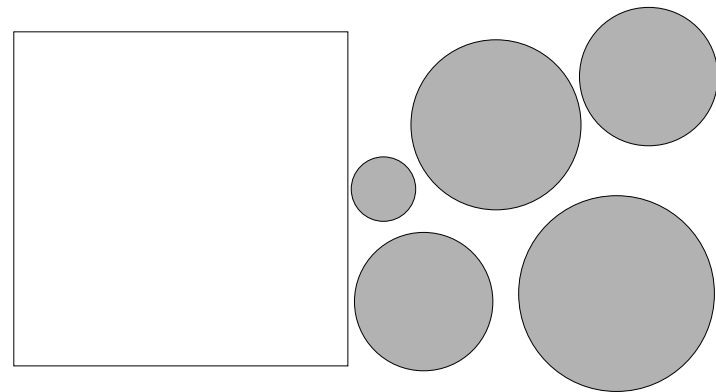
45 / 56

\$python nqueens.py

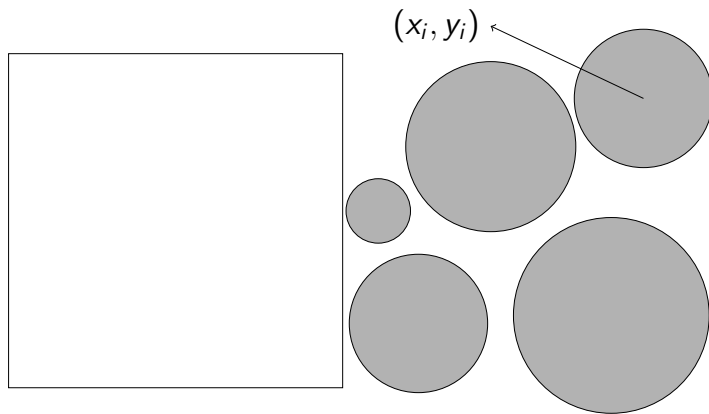
[q0(3), q1(1), q2(4), q3(7), q4(5), q5(0), q6(2), q7(6)]



## Problema de Empacotamento de Círculos

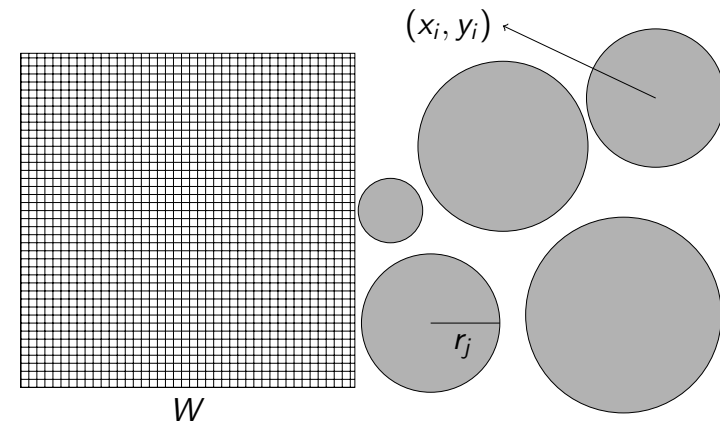


## Modelo em CP - Variáveis



49 / 56

## Modelo em CP - Domínio



50 / 56

## Modelo em CP - Domínio

- Variáveis  $x_i, y_i$  indicando onde um item  $i$  será empacotado.
- $Dom(x_i) = [r_i, \dots, W - r_i]$
- $Dom(y_i) = [r_i, \dots, W - r_i]$
- Para cada par de itens  $i, j$ :  $(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2$ .

```
from ortools.constraint_solver import pywrapcp
solver = pywrapcp.Solver("circle-packing")

W=4157
n=5
r = [1218, 1058, 861, 861, 401]

x = [solver.IntVar(r[i], W - r[i], "x%i" % i) for i in range(n)]
y = [solver.IntVar(r[i], W - r[i], "y%i" % i) for i in range(n)]

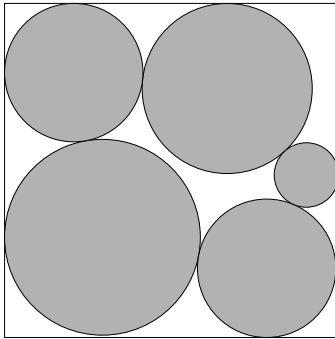
for i in range(n):
    for j in range(i):
        solver.Add(( x[i] - x[j])*(x[i] - x[j])
                    + (y[i] - y[j])*(y[i] - y[j])
                    >= (r[i] + r[j])*(r[i] + r[j]))

db = solver.Phase(x + y,
                  solver.INT_VAR_DEFAULT,
                  solver.INT_VALUE_DEFAULT)
solver.NewSearch(db)

solver.NextSolution()
print([(x[i], y[i]) for i in range(n)])
solver.EndSearch()
```

51 / 56

## Solução



53 / 56

## Conclusões

- Programação por Restrições em alguns casos é uma alternativa para programação linear inteira.
- É eficiente para vários problemas, em particular para problemas de alocação e escalonamento.
- É simples de implementar.

54 / 56

## Referências

- OR-Tools, <https://developers.google.com/optimization/>
- Hokama, P., Miyazawa, F. K., & Schouery, R. C. (2016). A bounded space algorithm for online circle packing. *Information Processing Letters*, 116(5), 337-342.
- Hokama, P. H. D. B. (2011). O problema do caixeiro viajante com restrições de empacotamento tridimensional.
- Rossi, F., Van Beek, P., & Walsh, T. (Eds.). (2006). *Handbook of constraint programming*. Elsevier.
- Barták, R. (2010). History of Constraint Programming. *Wiley Encyclopedia of Operations Research and Management Science*.
- Jean-Francois Puget (2014) Constraint Programming History, [https://www.ibm.com/developerworks/community/blogs/jfp/entry/constraint\\_programming\\_history?lang=en](https://www.ibm.com/developerworks/community/blogs/jfp/entry/constraint_programming_history?lang=en)

55 / 56

## Obrigado!

Pedro Henrique Del Bianco Hokama [hokama@unifei.edu.br](mailto:hokama@unifei.edu.br)

56 / 56