

Programación con Restricciones

Constraint Programming [MII-771]

Capítulo 1: Introducción

Dr. Ricardo Soto

[ricardo.soto@ucv.cl]

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

Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso

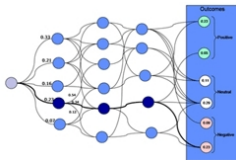


PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO



1. Introducción

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



```

1 package main{
2 import java.io.*
3 import java.net.*
4 import java.util.*
5
6 public class ServletFormaEstado implements
7     javax.servlet.http.HttpServlet
8     javax.servlet.http.HttpServletRequest
9     javax.servlet.http.HttpServletResponse
10 {
11     public void doGet(HttpServletRequest request, HttpServletResponse response)
12         throws ServletException, IOException
13     {
14         String nombreParametro = request.getParameter("nombre")
15         String telefonoParametro = request.getParameter("telefono")
16         String idClienteParametro = request.getParameter("id_cliente")
17
18         String nombreFormaEstado = request.getParameter("nombre")
19         String telefonoFormaEstado = request.getParameter("telefono")
20         String idClienteFormaEstado = request.getParameter("id_cliente")
21
22         String nombreFormaEstado = request.getParameter("nombre")
23         String telefonoFormaEstado = request.getParameter("telefono")
24         String idClienteFormaEstado = request.getParameter("id_cliente")
25
26         String nombreFormaEstado = request.getParameter("nombre")
27         String telefonoFormaEstado = request.getParameter("telefono")
28         String idClienteFormaEstado = request.getParameter("id_cliente")
29
30         String nombreFormaEstado = request.getParameter("nombre")
31         String telefonoFormaEstado = request.getParameter("telefono")
32         String idClienteFormaEstado = request.getParameter("id_cliente")
33
34         String nombreFormaEstado = request.getParameter("nombre")
35         String telefonoFormaEstado = request.getParameter("telefono")
36         String idClienteFormaEstado = request.getParameter("id_cliente")
37
38         String nombreFormaEstado = request.getParameter("nombre")
39         String telefonoFormaEstado = request.getParameter("telefono")
40         String idClienteFormaEstado = request.getParameter("id_cliente")
41
42         String nombreFormaEstado = request.getParameter("nombre")
43         String telefonoFormaEstado = request.getParameter("telefono")
44         String idClienteFormaEstado = request.getParameter("id_cliente")
45
46         String nombreFormaEstado = request.getParameter("nombre")
47         String telefonoFormaEstado = request.getParameter("telefono")
48         String idClienteFormaEstado = request.getParameter("id_cliente")
49
50         String nombreFormaEstado = request.getParameter("nombre")
51         String telefonoFormaEstado = request.getParameter("telefono")
52         String idClienteFormaEstado = request.getParameter("id_cliente")
53
54         String nombreFormaEstado = request.getParameter("nombre")
55         String telefonoFormaEstado = request.getParameter("telefono")
56         String idClienteFormaEstado = request.getParameter("id_cliente")
57
58         String nombreFormaEstado = request.getParameter("nombre")
59         String telefonoFormaEstado = request.getParameter("telefono")
60         String idClienteFormaEstado = request.getParameter("id_cliente")
61
62         String nombreFormaEstado = request.getParameter("nombre")
63         String telefonoFormaEstado = request.getParameter("telefono")
64         String idClienteFormaEstado = request.getParameter("id_cliente")
65
66         String nombreFormaEstado = request.getParameter("nombre")
67         String telefonoFormaEstado = request.getParameter("telefono")
68         String idClienteFormaEstado = request.getParameter("id_cliente")
69
70         String nombreFormaEstado = request.getParameter("nombre")
71         String telefonoFormaEstado = request.getParameter("telefono")
72         String idClienteFormaEstado = request.getParameter("id_cliente")
73
74         String nombreFormaEstado = request.getParameter("nombre")
75         String telefonoFormaEstado = request.getParameter("telefono")
76         String idClienteFormaEstado = request.getParameter("id_cliente")
77
78         String nombreFormaEstado = request.getParameter("nombre")
79         String telefonoFormaEstado = request.getParameter("telefono")
80         String idClienteFormaEstado = request.getParameter("id_cliente")
81
82         String nombreFormaEstado = request.getParameter("nombre")
83         String telefonoFormaEstado = request.getParameter("telefono")
84         String idClienteFormaEstado = request.getParameter("id_cliente")
85
86         String nombreFormaEstado = request.getParameter("nombre")
87         String telefonoFormaEstado = request.getParameter("telefono")
88         String idClienteFormaEstado = request.getParameter("id_cliente")
89
90         String nombreFormaEstado = request.getParameter("nombre")
91         String telefonoFormaEstado = request.getParameter("telefono")
92         String idClienteFormaEstado = request.getParameter("id_cliente")
93
94         String nombreFormaEstado = request.getParameter("nombre")
95         String telefonoFormaEstado = request.getParameter("telefono")
96         String idClienteFormaEstado = request.getParameter("id_cliente")
97
98         String nombreFormaEstado = request.getParameter("nombre")
99         String telefonoFormaEstado = request.getParameter("telefono")
100        String idClienteFormaEstado = request.getParameter("id_cliente")
101    }
102 }

```

Objetivo?

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

2. Ejemplos

Ejemplo 1

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

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

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

2. Ejemplos

Modelo - CSP (Constraint Satisfaction Problem)

- Variables

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

- Restricciones

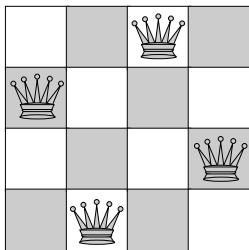
$$\begin{array}{rcccccc}
 & & 1000 \cdot S & + & 100 \cdot E & + & 10 \cdot N & + & D \\
 & & + & 1000 \cdot M & + & 100 \cdot O & + & 10 \cdot R & + & E \\
 = & 10000 \cdot M & + & 1000 \cdot O & + & 100 \cdot N & + & 10 \cdot E & + & Y
 \end{array}$$

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

2. Ejemplos

Ejemplo 2

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



2. Ejemplos

Modelo

- Variables

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

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

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

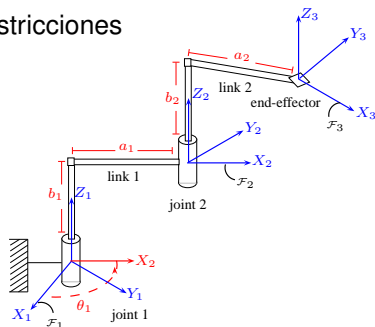
$$Q_i + i \neq Q_j + j \text{ (diagonal 1)}$$

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

2. Ejemplos

Ejemplos Reales

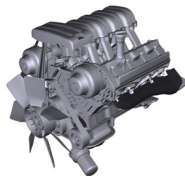
- Detección de errores de precisión en robots (IRCCYN Lab)
...+ de 500 variables y restricciones
- Diseño de un sistema de aire acondicionado para aviones (Dassault Aviation)
...+ de 1000 variables y restricciones



2. Ejemplos

+ Ejemplos Reales

- Instalación de antenas para asegurar un rango de cobertura mínima
- Diseño de ingeniería complejos (motores, mecanismos, estructuras...)
- Asignación de Tareas
- Transporte
- Biología molecular
- **Aplicable a cualquier problema que se pueda formular por medio de variables y restricciones**



3. Modelado

Solving = Modeling + Search

3. Modelado

Fases del modelado

- **Análisis** del problema
- **Formulación** del problema como un **CSP** (Constraint Satisfaction Problem)
- **Implementación** del modelo en un lenguaje para Constraint Programming

3. Modelado

Diversos Lenguajes para CP

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

3. Modelado

Principal ventaja

- **NO** es necesario especificar una **secuencia de pasos** a ejecutar para resolver el problema (programación imperativa)

...solamente se especifican las **propiedades** que debe tener la solución (restricciones)

3. Modelado

Ejemplo n-reinas en Zinc

```
...
int: n;
array [1..n] of var 1..n: board;

constraint
  forall (i in 1..n, j in i+1..n) (
    board[i]      != board[j]      /\
    board[i] + i != board[j] + j /\
    board[i] - i != board[j] - j;
  );
```

3. Modelado

Ejemplo n-reinas en C++

```

class ajedrez{
protected:
int a[8][8];
public:
int error;
ajedrez();
void limpiar();
void dibujar();
void insertar(int posx, int posy);
};

ajedrez::ajedrez(){
error=0;
for(int i=0;i<8;i++)
for(int j=0;j<8;j++)
a[i][j]=0;
}
void ajedrez::limpiar(){
for(int i=0;i<8;i++)
for(int j=0;j<8;j++)
a[i][j]=0;
}

```

```

void ajedrez::dibujar(){
for(int i=0;i<8;i++){
cout<<endl;
for(int j=0;j<8;j++){
cout<<a[i][j]<<" ";
}
}

void ajedrez::generar(int posx, int posy){
int x, y, i;
for(i=0;i<8;i++){
if(a[posx][i] == 1)error=1;

for(i=0;i<8;i++){
if(a[i][posy]== 1)error=1;
}
}

```

3. Modelado

```

or(i=0;i<8;i++){
if(a[i][posy]== 1)error=1;
}

x=posx;
y=posy;
while(x!= 0 && y!=0){
x--;
y--;
}
for(i=0;i<8;i++){
if((x+i)<8 & (y+i)<8){
if(a[x+i][y+i]==1 )error=1;

}
}
x=posx;
y=posy;
while(x!=0 ){
x--;
y++;
}

```

```

for(i=0;i<8;i++){
if((x+i<8 )&&((y-i>=0) && (y-i<8))){
if(a[x+i][y-i]==1)error=1;

}

}
a[posx][posy]=1;
}

int main()
{
int tablero[8][8],soluciones=0;
ajedrez *obj=new ajedrez();
for(int i1=0;i1<8;i1++){
for(int i2=0;i2<8;i2++){
for(int i3=0;i3<8;i3++){
for(int i4=0;i4<8;i4++){
for(int i5=0;i5<8;i5++){
for(int i6=0;i6<8;i6++){
for(int i7=0;i7<8;i7++){
for(int i8=0;i8<8;i8++){

```

3. Modelado

```

obj->insertar(0,i1) ;
obj->insertar(1,i2) ;
obj->insertar(2,i3) ;
obj->insertar(3,i4) ;
obj->insertar(4,i5) ;
obj->insertar(5,i6) ;
obj->insertar(6,i7) ;
obj->insertar(7,i8) ;
if(obj->error){
obj->limpiar();
obj->error=0; } else
{
clrscr();
obj->dibujar();
soluciones++;

```

```

cout<<"Solucion " <<soluciones<<endl;
cout<<"Las coordenadas fueron: " <<endl;
cout<<"1 - " <<i1+1<<endl;
cout<<"2 - " <<i2+1<<endl;
cout<<"3 - " <<i3+1<<endl;
cout<<"4 - " <<i4+1<<endl;
cout<<"5 - " <<i5+1<<endl;
cout<<"6 - " <<i6+1<<endl;
cout<<"7 - " <<i7+1<<endl;
cout<<"8 - " <<i8+1<<endl;
obj->limpiar();
getch();
}

}}}}}}}}
delete obj;
getch();
return 0;
}

```


4. Optimización con restricciones

Constraint Programming se puede extender fácilmente para resolver problemas de optimización

- CSP is about satisfaction
- COP is about optimization

4. Optimización con restricciones

Modelado

- Basta con agregar una función objetivo

```
a in [1,6]
```

```
b in [3,4]
```

```
a>b
```

```
minimizar: a+b
```

4. Optimización con restricciones

Ejemplo 2

- Considere una fábrica que debe satisfacer una determinada demanda de productos.
- Estos productos pueden ser **manufacturados internamente** o comprados en un **mercado externo**.



Objetivo?

Determinar la cantidad de productos que se deben **producir internamente** y los que se deben **comprar** de manera tal de **minimizar** el costo total.

4. Optimización con restricciones

```
...
int capacity[resourceList];
Product productSet[productList];

//constraint noExceedCapacity
forall(r in resourceList)
    capacity[r] >= sum(p in productList)
                    (productSet[p].consumption[r] *
                     productSet[p].inside);

//constraint satisfyDemand
forall(p in productList)
    productSet[p].inside + productSet[p].outside >=
    productSet[p].demand;

//constraint minimizeCost
[minimize]
    sum(p in productList)
        (productSet[p].insideCost * productSet[p].inside +
         productSet[p].outsideCost * productSet[p].outside);
...
```

5. Comparación con otras técnicas de optimización

Metaheurísticas

- Idea principal: encontrar una solución suficientemente buena en un período de tiempo limitado
- Es necesario implementar un algoritmo de resolución específico al problema en cuestión

en CP...

- **NO** es necesario implementar un algoritmo de resolución específico al problema en cuestión.
- Búsqueda global: en un problema de gran envergadura podría requerir de tiempos de resolución excesivos...

6. Desafíos

Modelado

- Lenguajes simples de utilizar
- Facilitar las tareas de modelado

Búsqueda

- Mejorar tiempos de resolución
- Explicar causas de las inconsistencias
- Análisis del comportamiento de heurísticas

7. Entidades fuertemente involucradas en CP

- IBM
- Microsoft Research
- University College Cork, Ireland
- University of Melbourne, Australia
- University of Singapore, Singapore
- University of Nantes, France
- University of York, UK
- University of Waterloo, Canada
- Brown University, USA
- University of California, Irvine, USA
- University of Padova, Italy
- University of Nice Sophia Antipolis, France
- University of Hong Kong, China
- Royal Institute of Technology, Sweden ...500+

8. Conferencias

- AAI (American Association for Artificial Intelligence)
- IJCAI (International Joint Conferences on Artificial Intelligence)
- ICLP (International conference on Constraint Logic Programming)
- CP (Principles and practice of Constraint Programming)
- PPDP (Principles and practice of Declarative programming)
- ICTAI (International conference on Tools with Artificial Intelligence)
- SAC (Symposium on Applied Computing)
- SARA (Symposium on Abstraction, Reformulation and Approximation)
- PADL (Practical Aspects of Declarative Languages)
- ECAI (European Conference on Artificial Intelligence)
- CPAIOR (Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems)
- ...20+

9. Journals y revistas

- ACM TOPLAS
- ACM TOCL
- Constraints
- AIEDAM
- SIAM J. Scientific Computing
- Reliable Computing
- ...10+