

Solución de Problemas con CCP

Introducción

slides basados en el curso “constraint Programming” de
Christian Schulte ²
Profesor: Camilo Rueda ¹

¹Universidad Javeriana-Cali,

²KTH Royal Institute of Technology, Sweden

PUJ 2008

Qué es Programación de Restricciones?

Metodología, formalismo y técnicas
para resolver cierta clase de problemas

Modelo de restricciones para GCP

- Variables:

$G, A, T, C, O, M, E, P, L \in \{0, \dots, 9\}$

- Restricciones:

$\text{distinct}(G, A, T, C, O, M, E, P, L)$

$$\begin{aligned}
 & 1000 \times G + 100 \times A + 10 \times T + A \\
 + & 1000 \times C + 100 \times O + 10 \times M + E \\
 = & 10000 \times P + 1000 \times O + 100 \times L + 10 \times L + O \\
 & G \neq 0 \quad P \neq 0
 \end{aligned}$$

Resolver GCP

- Encontrar valores para las variables

Tales que

todas las restricciones se satisfacen

Encontrar una solución

- Calcular con los valores **posibles**
 - en lugar de **enumerar todas las asignaciones**
- **Podar** valores inconsistentes
 - **propagación de restricciones**
- **Búsqueda**
 - **ramificar:** definir el árbol de búsqueda
 - **explorar:** buscar solución en el árbol de búsqueda

Propagación

Conceptos importantes:

- El **store** de restricciones
- **Propagador**
- Propagación de restricciones

El store

ejemplo

$$x \in \{3, 4, 5\} \quad y \in \{3, 4, 5\}$$

- Relaciona variables con sus valores posibles
 - guarda restricciones **básicas**

El store

ejemplo



$$x \in \{3, 4, 5\}$$

$$y \in \{3, 4, 5\}$$

- Relaciona variables con sus valores posibles
 - guarda restricciones **básicas**

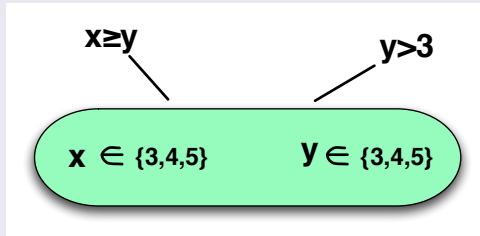
Propagadores

- Implementan restricciones **no** básicas

$\text{distinct}(x_1, \dots, x_n)$

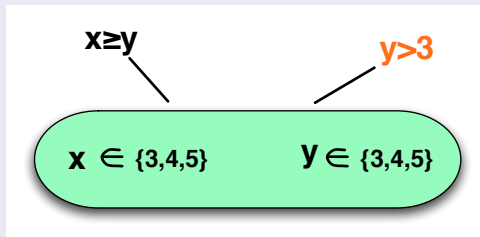
$$z + 3 \times y = w$$

Propagadores



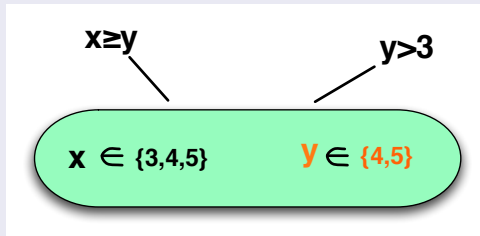
- amplifican el store mediante propagación de restricciones

Propagadores



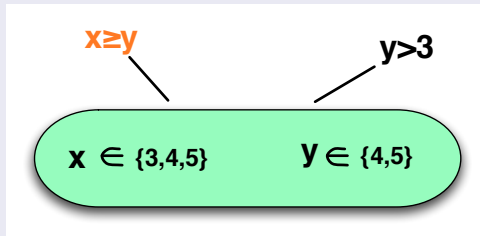
- amplifican el store mediante propagación de restricciones

Propagadores



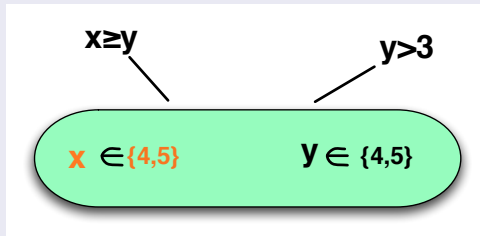
- amplifican el store mediante propagación de restricciones

Propagadores



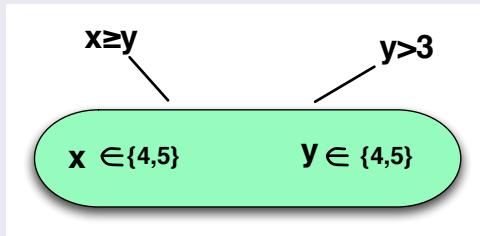
- amplifican el store mediante propagación de restricciones

Propagadores



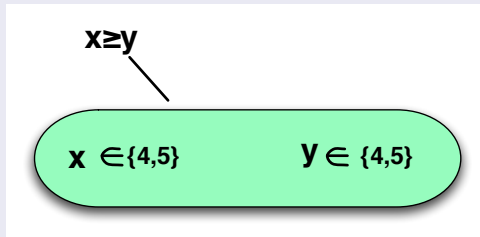
- amplifican el store mediante propagación de restricciones

Propagadores



- amplifican el store mediante propagación de restricciones

Propagadores



- amplifican el store mediante propagación de restricciones

Propagación para GCP

- Resultados en el store:

$A \in \{0, 2, 3, \dots, 9\}$	$C \in \{0, 2, 3, \dots, 9\}$	$E \in \{0, 2, 3, \dots, 9\}$
$G \in \{2, 3, \dots, 9\}$	$L \in \{0, 2, 3, \dots, 9\}$	$M \in \{0, 2, 3, \dots, 9\}$
$O \in \{0, 2, 3, \dots, 9\}$	$P = 1$	$T \in \{0, 2, 3, \dots, 9\}$

- La propagación sola **no basta!**
 - crear subproblemas más sencillos
 - **ramificar**

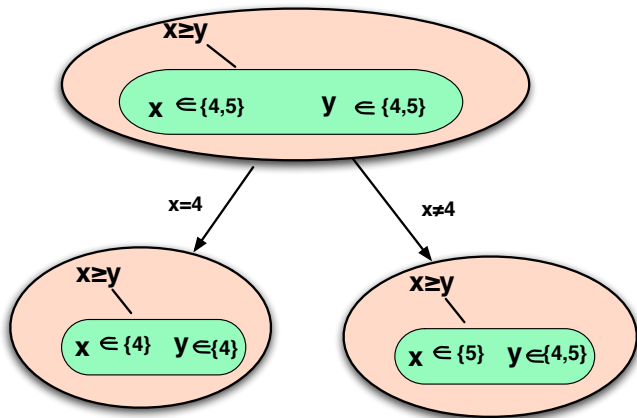
Restricciones y Propagadores

- Las restricciones establecen **relaciones** entre variables
 - Qué combinaciones de valores satisface la restricción
- Los propagadores **implementan** restricciones
 - filtrar valores en conflicto con la restricción
- La propagación de restricciones opera los propagadores para varias restricciones

Búsqueda

- Conceptos importantes
 - Ramificación
 - Exploración
 - Heurísticos de ramificación
 - búsqueda de la mejor solución

Búsqueda: ramificación

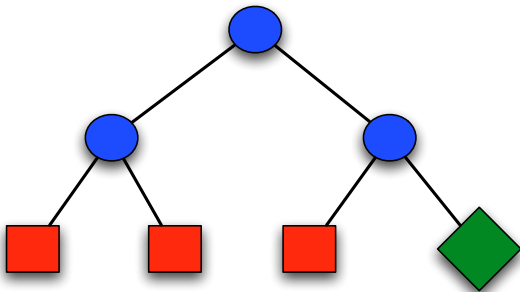


- crear subproblemas con información adicional

Ejemplo: estrategia de ramificación

- tomar variable x con al menos dos valores
- toar valor n del dominio de x
- ramificar con
 $x = n$ y con $x \neq n$
- es parte del modelo

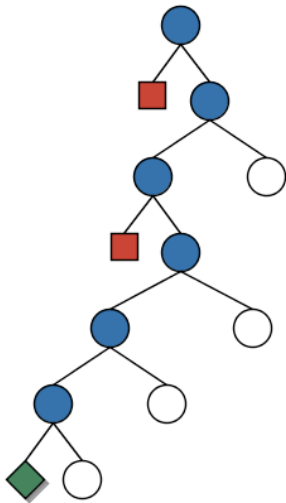
Búsqueda: exploración



- iterar propagación y ramificación
- ortogonal: ramificación \leftrightarrow exploración
- nodos

fallido no resuelto Exitoso

Exploración en GCP



		G	A	T	A
+		C	O	M	E
=	P	O	L	L	O
		3	4	0	4
+		9	2	5	8
=	1	2	6	6	2

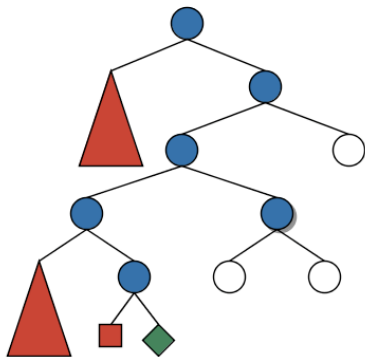
Heurísticas para ramificación

- Qué variable
 - la de menores valores posibles (“first-fail”)
 - heurístico dependiente de la aplicación
- qué valor
 - mínimo, mediana, máximo
 - $x = m$ o $x \neq m$
 - partir con la mediana m
 - $x < m$ o $x \geq m$
- Específico al problema

Optimización: búsqueda de la mejor solución

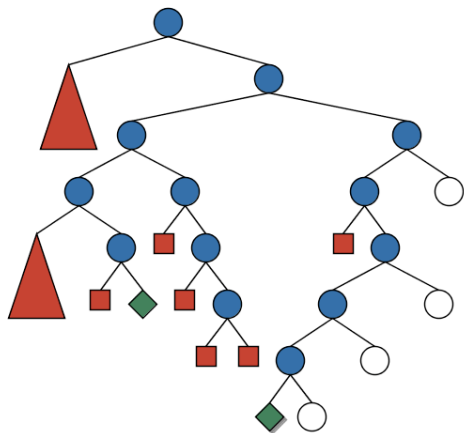
- Estrategia ingenua
 - calcule todas las soluciones
 - escoja la mejor
- Estrategia “ramificar y acotar” (“branch-and-bound”)
 - calcule la primera solución
 - agregue restricción de “mejora” para abrir los nodos
 - la siguiente solución ser mejor
 - poda el espacio de búsqueda

Optimización: problema DTA



- Explore con restricciones adicionales

Optimización: problema DTA



= 1 2 7 8 0

- Garantiza mejores soluciones

Modelar DTA

- Restricciones y ramificación, como en GCP
- Orden entre soluciones con restricciones
 - la mejor solución hasta ahora D, E, J, A, T, O, N, S
 - nodo actual D, E, J, A, T, O, N, S
 - restricción que se agrega:
$$10000 \times A + 1000 \times N + 100 \times T + 10 \times E + S$$
$$<$$
$$10000 \times A + 1000 \times N + 100 \times T + 10 \times E + S$$

Gran número de aplicaciones

- Tempotabulación
- scheduling
- planeamiento de equipos
- asignación de recursos
- optimización
- asignación de muelles en aeropuertos
- scheduling de eventos deportivos
- composición musical
- secuenciamiento de genoma
- asignación de frecuencias
-

Basado en varias técnicas

- Inteligencia artificial
 - ideas básicas, búsqueda, ...
- investigación de operaciones
 - scheduling, flujos en redes,...
- algoritmos
 - grafos, “matching”, redes, ...
- Lenguajes de programación
 - programabilidad, extensibilidad, ...

Aspecto esencial

- “middleware” composicional para combinar
 - buenos algoritmos
 - subestructuras de problemas
- componentes (propagadores)
 - scheduling
 - grafos
 - secuencias
 - ...

Principios

- modelos para propagación de restricciones
 - propiedades y sus garantías
- propagación de restricciones fuerte
 - restricciones globales con métodos algorítmicos sólidos
 - principio de base: **la búsqueda mata**, ...
- Estrategias de ramificación
- estrategias de exploración

Area: scheduling de recursos

- Tareas
 - duración
 - recurso
- restricciones de precedencia
 - determinan el orden entre tareas
- restricciones de recurso
 - a lo sumo una tarea por recurso
(scheduling disyuntivo, no interruptible)

Scheduling: solución

- Tiempo de comienzo de cada tarea
- Todas las restricciones se satisfacen
- Tiempo de terminación lo más temprano posible

Scheduling: modelo

- Variable para tiempo de comienzo de la tarea a
 $start(a)$
- Restricción de precedencia: a antes que b
 $start(a) + dur(a) \leq start(b)$

Propagar precedencia

a antes-que *b*

$$\begin{array}{ll} \textit{start}(a) \in \{0, \dots, 7\} & \textit{dur}(a) = 1 \\ \textit{start}(b) \in \{0, \dots, 5\} & \textit{dur}(a) = 2 \end{array}$$

Propagar precedencia

a antes-que *b*

$$\begin{array}{lll} \text{start}(a) \in \{0, \dots, 7\} & \text{dur}(a) = 1 & \Rightarrow \text{start}(a) \in \{0, \dots, 2\} \\ \text{start}(b) \in \{0, \dots, 5\} & \text{dur}(a) = 2 & \Rightarrow \text{start}(b) \in \{3, \dots, 5\} \end{array}$$

Scheduling: modelo (2)

- Variable para tiempo de comienzo de la tarea a
 $start(a)$
- Restricción de precedencia: a antes que b
 $start(a) + dur(a) \leq start(b)$
- Restricción de recurso
 a antes que b
o bien
 b antes que a

Scheduling: modelo (2)

- Variable para tiempo de comienzo de la tarea a
 $start(a)$
- Restricción de precedencia: a antes que b
 $start(a) + dur(a) \leq start(b)$
- Restricción de recurso
 $start(a) + dur(a) \leq start(b)$
o bien
 b antes que a

Scheduling: modelo (2)

- Variable para tiempo de comienzo de la tarea a
 $start(a)$
- Restricción de precedencia: a antes que b
 $start(a) + dur(a) \leq start(b)$
- Restricción de recurso
 $start(a) + dur(a) \leq start(b)$
o bien
 $start(b) + dur(b) \leq start(a)$

Restricciones reificadas

- Usar variable de control $b \in \{0, 1\}$

$$c \Leftrightarrow b = 1$$

- Propagar

- c se cumple \Rightarrow propagar $b = 1$
- $\neg c$ se cumple \Rightarrow propagar $b = 0$
- $b = 1$ se cumple \Rightarrow propagar c
- $b = 0$ se cumple \Rightarrow propagar $\neg c$

Restricciones reificadas

- Usar variable de control $b \in \{0, 1\}$

$$c \Leftrightarrow b = 1$$

- Propagar

- c se cumple \Rightarrow propagar $b = 1$
- $\neg c$ se cumple \Rightarrow propagar $b = 0$
- $b = 1$ se cumple \Rightarrow propagar c
- $b = 0$ se cumple \Rightarrow propagar $\neg c$ ← difícil !!

Reificación para disyunción

- Reificar cada precedencia

$$[start(a) + dur(a) \leq start(b)] \leftrightarrow b_0 = 1$$

y

$$[start(b) + dur(b) \leq start(a)] \leftrightarrow b_1 = 1$$

- Modelar disyunción

$$b_0 + b_1 \geq 1$$