
Algoritmos Exactos y Metaheurísticas

Primer Semestre 2025

Universidad Diego Portales
Prof. Víctor Reyes Rodríguez

Objetivos

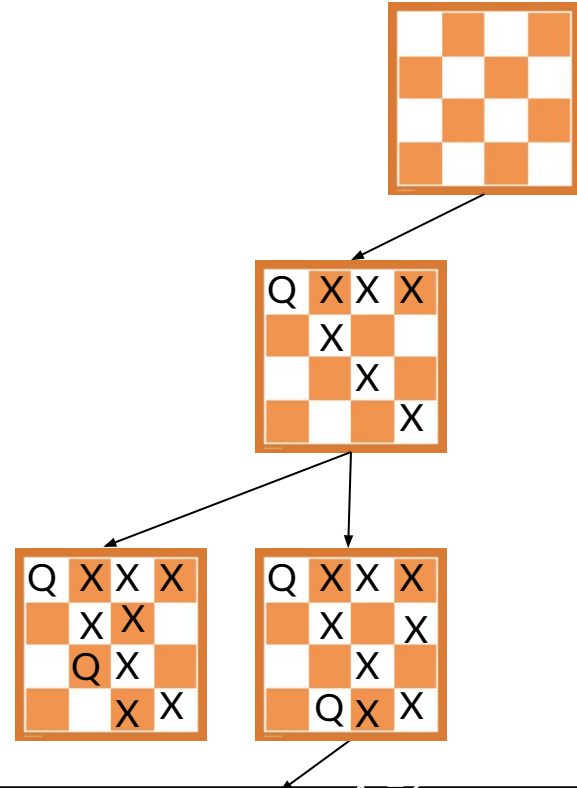
- Conocer y analizar las Técnicas Look-Ahead
 - Comprender la importancia de las heurísticas de selección de variable.
-

Forward Checking

- Pertenece a un grupo de técnicas clasificadas como Técnicas Look-Ahead.
 - Se basa en la idea de mirar hacia adelante en el árbol de búsqueda, para ver si al hacer una instanciación hace imposible asignarle un valor a otra variable no instanciada.
 - Veamos cómo se aplicaría usando el ejemplo de las N-Reinas...
-

Forward checking

- Disminuye el trashing.
- A pesar de que podrían existir menos nodos en el árbol, se podrían eventualmente realizar más chequeos, en comparación a las otras técnicas estudiadas.
- Continuar con el ejemplo...



Podríamos resumir lo anterior en los siguientes pasos:

1. Seleccionar x_i
 2. Instanciar (x_i, a_i) : $a_i \in D_i$
 3. Razonar hacia adelante: Eliminar de los dominios de las variables aún no instanciadas, aquellos valores inconsistentes con respecto a la instanciación (x_i, a_i) , de acuerdo al conjunto de restricciones.
 4. Si quedan valores posibles en los dominios de todas las variables por instanciar, entonces:
 - a. Si $i < n$, incrementar i , e ir al paso 1.
 - b. Si $i = n$, parar devolviendo la solución.
 5. Si existe una variable por instanciar sin valores posibles en su dominio entonces retractar los efectos de la asignación (x_i, a_i) :
 - a. Si quedan valores por intentar en D_i , ir al paso 2.
 - b. Si no quedan valores:
 - i. Si $i > 1$, decrementar i y volver al paso 2.
 - ii. Si $i = 1$, salir sin solución.
-

Forward Checking

- Se podría combinar con otros algoritmos. Por ejemplo con CBJ (el de la clase pasada).
→ Reducimos dominios hacia adelante, si llegamos a un deadend, hacemos el salto a la variable con el conflicto.
 - ¿Es siempre FC mejor que un BT simple?
 - Existen otras variantes de forward checking, por ejemplo:
 - Minimal Forward Checking (MFC)
-

MFC

- Es conocido también por “Lazy forward checking”
 - Revisa hacia adelante al igual que FC, sin embargo al encontrar un valor asignación factible para una variable pasa inmediatamente a la otra.
 - De esta manera, eventualmente, podríamos detectar vacíos de dominio (al igual que FC), pero sin realizar tantos chequeos.
 - Sigue siendo completo
-

Orden de instanciación

- El orden en el cual son instanciadas las variables afecta el tamaño del árbol de búsqueda → desempeño del algoritmo de búsqueda.
 - Heurísticas de selección de variable: Procedimiento generalmente *no costoso* computacionalmente, el cual tiene por objetivo seleccionar la siguiente variable a ser instanciada.
 - En general están basadas en la premisa fail-first “Para tener éxito, se debe intentar primero en donde más probablemente se fallará”
-

Heurísticas de selección de variable

- Generalmente entre más información utilicemos, mejor funcionará—> más costo de CPU.
 - Algunos ejemplos (generales):
 - Dom: Se selecciona la de menor dominio.
 - Dom+Deg: Lo mismo que dom, sin embargo, si hay empate se elige aquella que aparece en más restricciones.
 - Dom/Deg: Se selecciona aquella que minimiza el cuociente dom/deg.
 - Dependiente del problema
 - Su propuesta...
-

Técnicas de preproceso: Consistencia de arcos y nodos

- Antes de ver esto, es necesario saber ciertas propiedades de CSP/COP (se pueden demostrar, pero no lo veremos en este curso):
 - Todo CSP/COP puede ser transformado a un conjunto de restricciones binarias y unarias.
 - Todo CSP/COP con restricciones binarias y unarias puede ser modelado a través de un grafo.
 - Con lo anterior, podemos aplicar técnicas de preproceso para CSP/COP: Nodo consistencia, Arco consistencia, Camino consistencia.
-

Resumen

- Estudiamos un tipo de técnica que mira hacia adelante.
 - Próxima clase: Otros tipos de búsqueda con árboles (beam-search, A^* , MCTS)
-