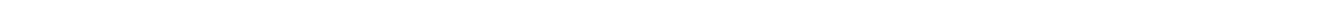

Algoritmos Exactos y Metaheurísticas

Primer Semestre 2025

Universidad Diego Portales
Prof. Víctor Reyes Rodríguez

Objetivos

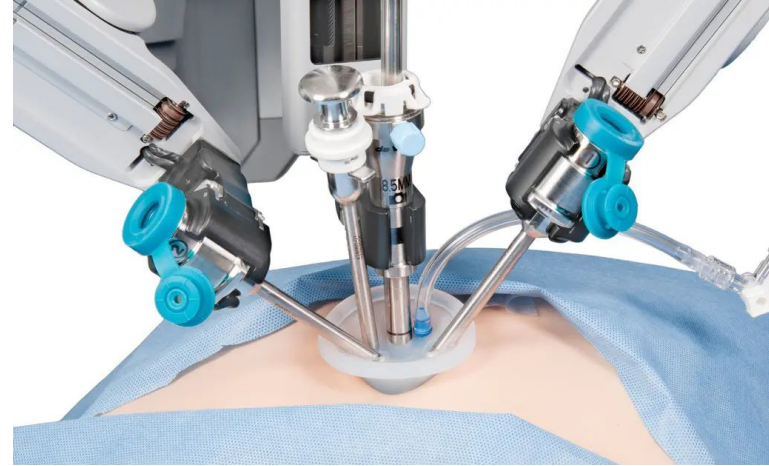
- Conocer otros algoritmos basados en árboles



Otras técnicas basadas en árboles

- Algunas de ellas no clasifican como búsquedas completas.
 - Revisaremos cuatro: Branch & Bound en problemas continuos, Beam-Search, Monte Carlo Tree Search y el famoso A^* (A-Star).
-

Problemas continuos



Elementos de los Problemas

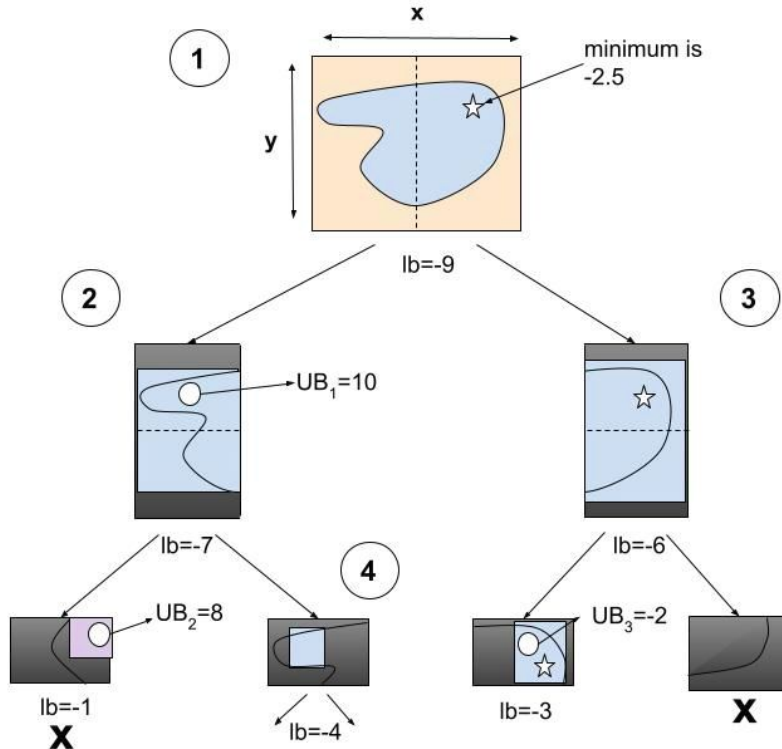
```
1 constants
2
3
4 variables
5 x1 in [0.001, 1.e3];
6 x2 in [0.001, 1.e3];
7 x3 in [0.001, 1.e3];
8 x4 in [0.001, 1.e3];
9 x5 in [0.001, 1.e3];
10 x6 in [0.001, 1.e3];
11 x7 in [0.001, 1.e3];
12 x8 in [0.001, 1.e3];
13 x9 in [0.001, 1.e3];
14 x10 in [0.001, 1.e3];
15 x11 in [0.01, 1.e3];
16
17 minimize x1*(ln(x1) - 6.05576803624071) + x2*(ln(x2) -
18 17.1307680362407) + x3*(ln(x3) - 34.0207680362407) + x4*(
19 ln(x4) - 5.88076803624071) + x5*(ln(x5) -
20 24.6877680362407) + x6*(ln(x6) - 14.9527680362407) + x7*(
21 ln(x7) - 24.0667680362407) + x8*(ln(x8) -
22 10.6747680362407) + x9*(ln(x9) - 26.6287680362407) + x10*(
23 ln(x10) - 22.1447680362407) - x11*ln(x11);
24
25 constraints
26 x1 + 2*x2 + 2*x3 + x6 + x10 - 2 =0;
27
28
29 x4 + 2*x5 + x6 + x7 - 1 =0;
30
31
32 x3 + x7 + x8 + 2*x9 + x10 -1 =0;
33
34
35 - x1 - x2 - x3 - x4 - x5 - x6 - x7 - x8 - x9 - x10 + x11 =0;
36
37
38 end
```

- Variables y dominios
- Función objetivo
- Una o más restricciones

En busca del óptimo global

- Necesito una técnica completa → Búsqueda a través de árboles.
 - No puedo hacer asignaciones a variables!! Los dominios son continuos...
 - Debo tomar en cuenta las aproximaciones de punto flotante.
-

Interval Branch & Bound



- Técnica basada en árboles de búsqueda.
- Toma en cuenta las aproximaciones de punto flotante realizadas por la máquina. Esto se logra usando una aritmética especial
- Al final de la búsqueda encuentra, en caso de existir, el óptimo global contenido en un intervalo de precisión arbitraria.

Bisección

- Consiste en dividir el dominio de algunas de las variables por la mitad, creando dos nuevos nodos.
 - Hasta la fecha, solo han sido creados métodos heurísticos, es decir, ven información parcial del problema para elegir la variable a bisectar.
 - Largest-First: Dominio más grande
 - Round-Robin: Van en orden
 - Smear-based: Miden el impacto de las variables en el problema
-

Filtrado

- Consiste en eliminar desde los bordes de los intervalos, valores que no son parte de ninguna solución.
 - También, se pueden eliminar nodos que con seguridad no nos ayudarán a encontrar un mínimo más pequeño que el actual.
 - Son utilizados una colección de técnicas, algunas más “caras” (computacionalmente hablando) que otras.
-

Upper bounding

- Consiste en buscar dentro de cada nodo, soluciones factibles, es decir, mínimos locales.
 - En general, son utilizados métodos de búsqueda local debido al costo computacional.
-

Selección de Nodo

- El mismo nombre lo dice, corresponde a elegir alguno de los nodos del árbol y procesarlo con los métodos anteriormente explicados.
 - Al igual que la bisección, son utilizados métodos heurísticos, por ejemplo algo que se llama minLB
 - Mejora a lo anterior, FeasibleDiving (minLB + Búsqueda en profundidad)
-

¿Y qué se está haciendo en el área?

- Selección de Nodo: Técnicas de aprendizaje por refuerzo para seleccionar la heurística más adecuada (hiperheurísticas)
 - Upper bounding: Uso de técnicas iterativas de búsqueda local y metaheurísticas como: Simulated Annealing, PSO y paralelización con threads.
 - Restarts con aprendizaje no supervisado.
-

Software

- Ibex: Librería de código abierto implementado en C++. Contiene los métodos del estado del arte.



Beam-Search

- Técnica incompleta.
 - Puede verse como una adaptación de una búsqueda de árbol, en donde solo los nodos más prometedores se consideran en cada nivel del árbol.
 - Utiliza un parámetro w , el cual representa el número de “beams” que consideraremos.
 - La pregunta es: ¿Cómo elegir los w mejores?
-

Beam-Search

- Depende del problema. En el paper “*A beam search approach to the container loading problem*” se ejecuta un “greedy” (este concepto lo vamos a definir de manera precisa la próxima sesión) en cada uno de los w^2 nodos.
 - Dicho greedy llega a una solución. Con esto podemos evaluar que tan “prometedor” es nuestra solución parcial.
 - Utiliza además el concepto de doble esfuerzo: en cada paso se incrementa el w a $\sqrt{2} \cdot w$.
-

Monte Carlo Tree Search

- Técnica incompleta.
 - Utilizada en Juegos: Go, Ajedrez, juegos de arcade como donkey kong, pacman (caída carnet), juegos en tiempo real, problemas de optimización como transporte, scheduling entre muchos más.
 - En los últimos años se ha mezclado esta técnica con Reinforcement Learning - RL (Alpha Go)
-

Monte Carlo Tree Search

- Mantiene un árbol de estados, en donde cada nodo además contiene un “valor” que mide que tan bueno es el estado con respecto al objetivo.
 - Balance entre “Exploración” y “Explotación”.
 - Trabaja en base a 4 pasos: Selección, Expansión, Simulación y Propagación hacia atrás.
-

Monte Carlo Tree Search

- Selección: Se escoge una secuencia de pasos, para ser explorada aún más. La idea es mantener un equilibrio entre caminos poco seleccionados vs nodos con un gran valor para así obtener más información de movimientos prometedores

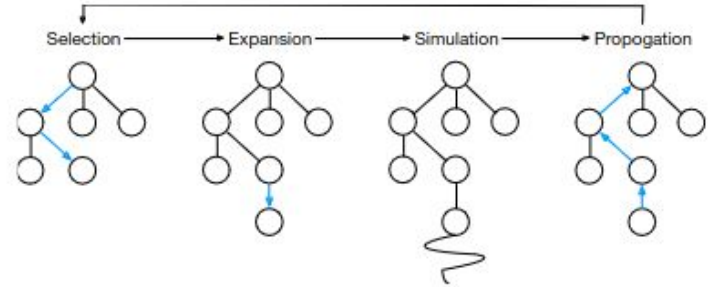


Figure 1. The steps for Monte Carlo Tree Search.

Monte Carlo Tree Search

- Expansión: Luego de elegir un nodo, se expande, realizando para esto una acción. En el MCTS clásico esta acción es random.
- Simulación: Se genera una simulación a partir de este nuevo nodo, hasta llegar a un nodo hoja. Se hace de manera aleatoria.

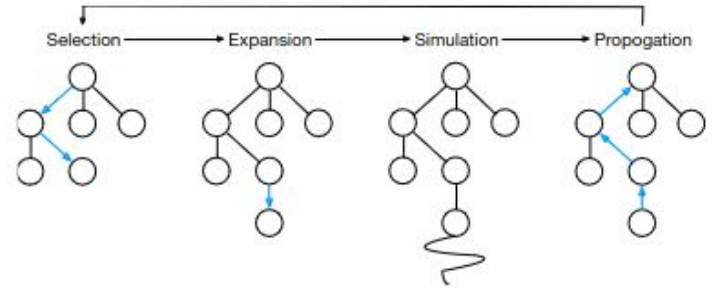


Figure 1. The steps for Monte Carlo Tree Search.

Monte Carlo Tree Search

- Propagación hacia atrás: Los resultados de la simulación se propagan hacia la secuencia de nodos.
- Como se mencionó anteriormente, una tendencia corresponde a utilizar una técnica de RL en la parte de la simulación.

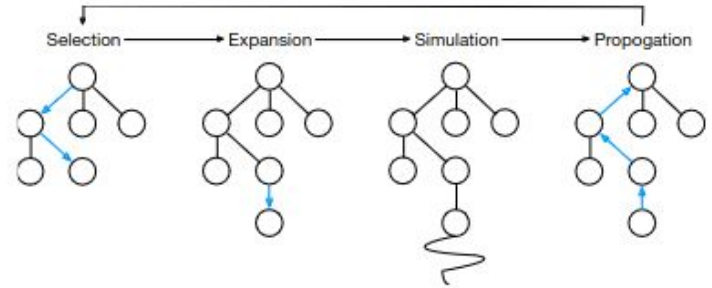


Figure 1. The steps for Monte Carlo Tree Search.

A* (A-estrella)

- Algoritmo legendario de IA.
 - Utilizado generalmente en planificación de caminos, usa una combinación entre heurísticas y búsqueda del menor camino.
 - Es un algoritmo del tipo “best-first”, pues cada nodo es evaluado a través de :
 $f(v)=h(v)+g(v)$.
 - h corresponde a la heurística admisible de distancia desde el nodo hasta el objetivo. No debe sobreestimar.
 - g corresponde a la distancia entre el nodo inicial y el camino realizado hasta dicho punto de la búsqueda.
-

Listo con técnicas completas....

- La próxima clase empezamos con técnicas incompletas
-