Optimización Discreta

Alan Reyes-Figueroa

Modelación y Simulación (Aula 15) 19.agosto.2025

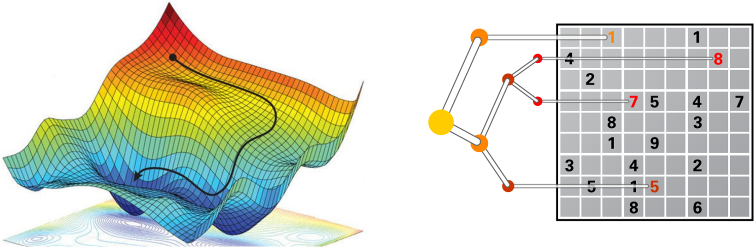
Optimización Discreta

Recordemos el problema de optimización

$$\text{Hallar } \mathbf{x}^* = \text{argmin}_{\mathbf{x}\in\Omega} F(\mathbf{x}), \qquad (\text{sujeto a } G(\mathbf{x})).$$

Separamos nuestras estrategias de búsqueda en 2 grandes grupos:

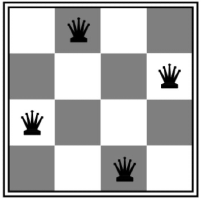
- Optimización continua,
- Optimización combinatoria o discreta.

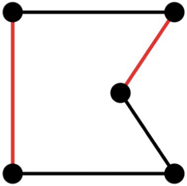


(a) Optimización continua, (b) Optimización discreta.

En el caso discreto, el espacio de estados = espacio de configuraciones Ω corresponde a un conjunto finito (muchas veces excesivamente grande).
La mayoría de problemas de optimización combinatoria se reduce a

- Hallar la(s) configuración(es) que cumpla ciertas restricciones (e.g. N-queens, Scheduling, Assignment Problem),
- Hallar la(s) configuración(es) óptima(s) dentro del espacio de configuraciones (e.g. TSP, Knapsack, SAT).





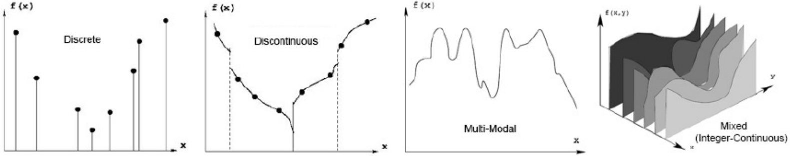
Optimización Discreta

Introducimos ahora algunos métodos del área llamada optimización estocástica. La idea principal es que desarrollamos métodos de búsqueda en donde interviene algunas de las siguientes:

- aleatoriedad,
- distribuciones.

Esto métodos sirven principalmente para atacar problemas de tipo

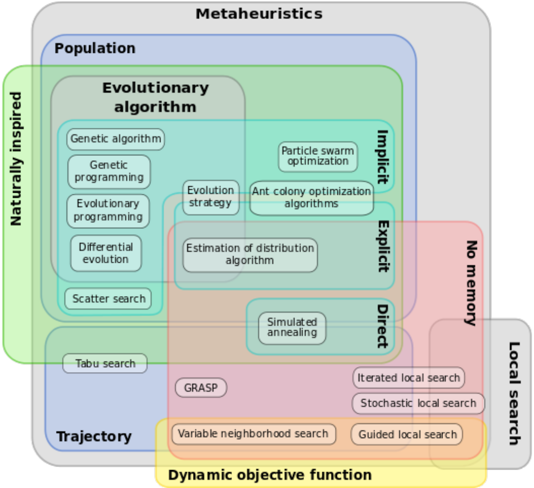
- espacio factible discreto
- múltiples mínimos locales (multimodal)



Algunas funciones problemáticas: (a) discreta, (b) discontinua, (c) multimodal, (d) mixta.

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Heurísticas y Metaheurísticas



UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Representación

En optimización combinatoria, tenemos varios tipos de representaciones:

Binaria:

$$\mathbf{x} = \boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{1} \in \{0,1\}^n.$$

Aquí $x_i \in \{0,1\}$. El espacio de configuraciones es $\Omega = \{0,1\}^n$ y $|\Omega| = 2^n$.

Entera:

$$\mathbf{x} = \boxed{5}\boxed{3}\boxed{0}\boxed{1}\boxed{2}\boxed{2}\boxed{4}\boxed{3}\boxed{7}\boxed{2} \in \mathbb{Z}^n$$

Aquí las $x_i \in \mathbb{Z}$, y el espacio de configuraciones es $\Omega = \mathbb{Z}^n$, en el caso no acotado. En el caso semiacotado, usualmente tenemos $x_i \geq 0$ y $\Omega = \mathbb{N}^n$. También podemos tener el caso acotado, cuando $a \leq x_i \leq b$. Aquí $\Omega = [a..b]^n$ y $|\Omega| = (b-a+1)^n$.

Continua o Float:

$$\mathbf{x} = \boxed{3.1}\boxed{2.5}\boxed{4.33}\boxed{-1.1}\boxed{0.15}\boxed{2.7}\boxed{5.0}\boxed{4.4}\boxed{-0.6}\boxed{2.6} \in \mathbb{R}^n$$

Aquí las $x_i \in \mathbb{R}$. El espacio de configuraciones es $\Omega = \mathbb{R}^n$. En el caso acotado, tenemos que $a \leq x_i \leq b$.

Representación

Un caso particular de la representación entera es la siguiente:

Permutaciones:

$$\mathbf{x} = \boxed{2}\boxed{6}\boxed{3}\boxed{9}\boxed{8}\boxed{0}\boxed{4}\boxed{1}\boxed{7}\boxed{5} \in S_n.$$

Aquí el vector **x** es un reordenamiento del conjunto $\{0, 1, 2, \ldots, n-1\}$. Deben aparecer todos los valores del conjunto, una vez cada uno.
En este caso el espacio de configuraciones es $\Omega = S_n$ y $|\Omega| = n!$.

Importante! Los algoritmos de optimización discreta son bastante sensibles al tipo de representación que usamos en nuestro problema.

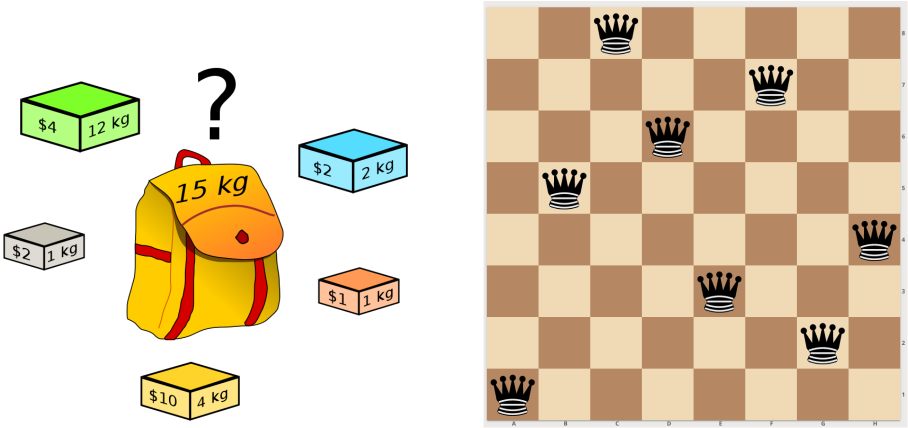
Ejemplos

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

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

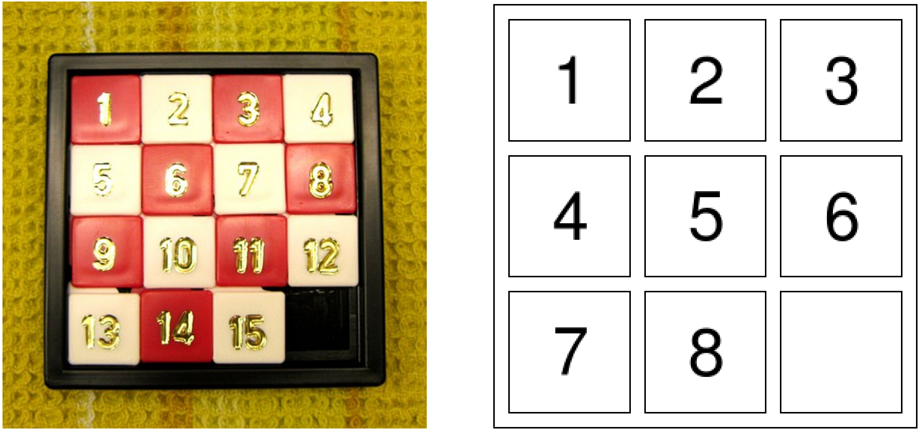
Sudoku.

Ejemplos



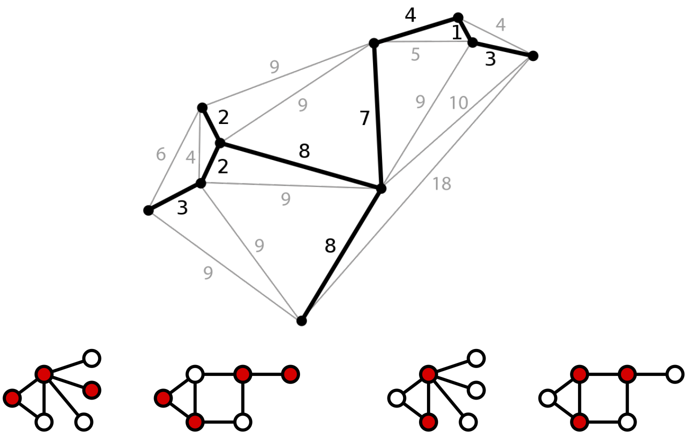
(a) *Knapsack problem*, (b) *8-queens Problem*.

Ejemplos

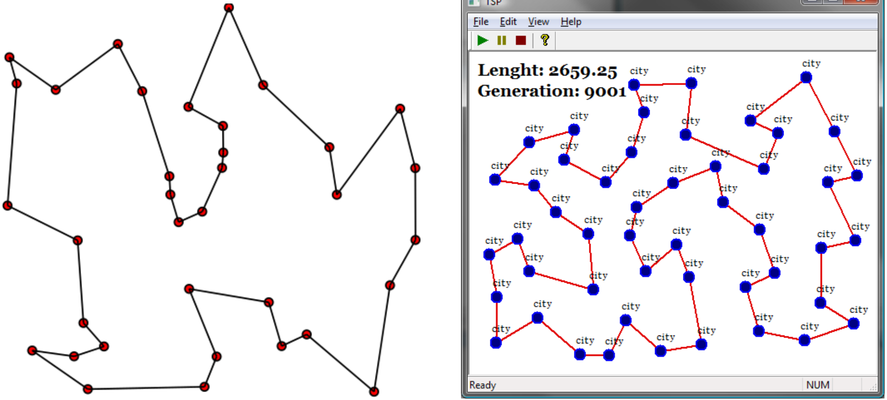


(a) 15-puzzle, (b) 8-puzzle.

Ejemplos



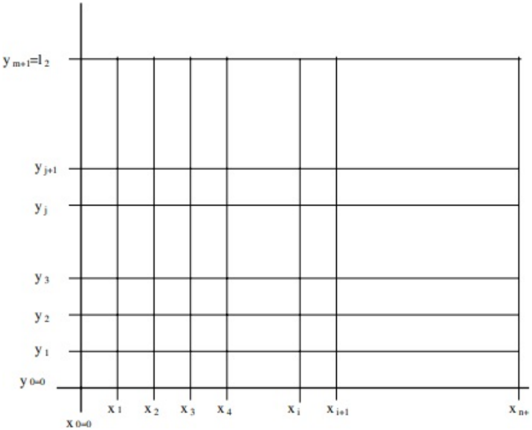
(a) *Minimum spanning tree*, (b) *Vertex covering*.



Problema del TSP.

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Ejemplos



Grid de discretización de un dominio continuo.

Búsqueda Parcial y Soluciones Subóptimas

En optimización discreta o combinatoria, es común tratar de buscar soluciones en espacios altamente grandes.
En un problema de optimización discreta, la idea es hallar la configuración óptima $\mathbf{x}^*$ de

$$\text{Hallar } \mathbf{x}^* = \text{argmin}_{\mathbf{x}\in\Omega} F(\mathbf{x})$$

sin tener que hacer una **búsqueda exhaustiva** (1 por 1). (Esto incurre casi siempre en un problema NP o NP-hard).

La idea es desarrollar algoritmos "inteligentes", en el sentido que no exploran todo el espacio de configuraciones Ω de forma completa.
En lugar de ello, la mayoría de métodos hace una **búsqueda simplificada** o **aleatoria**, pero de forma inteligente. En esta exploración parcial de Ω, casi siempre obtienen una solución **subóptima**, pero aceptable del problema.