

Búsqueda local

L. Mandow

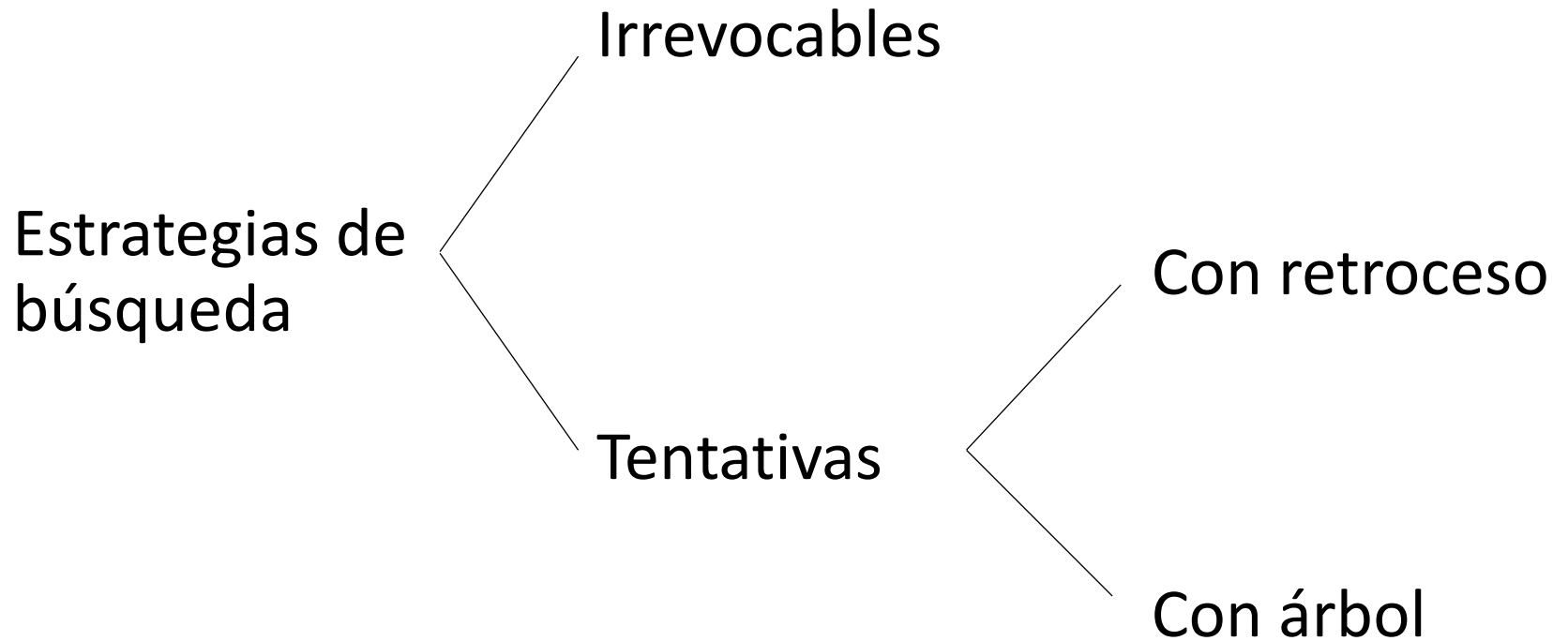
Dpto. Lenguajes y Ciencias de la Computación

Universidad de Málaga

Abril, 2020

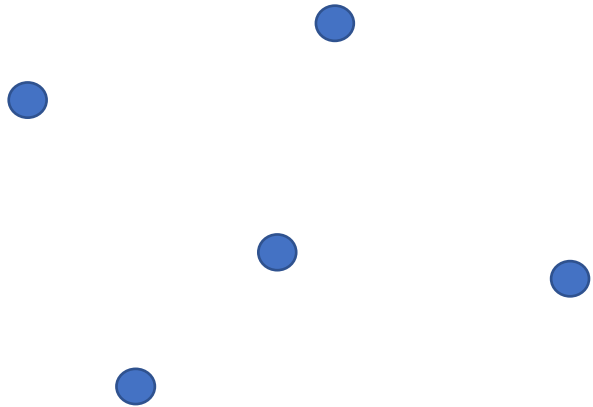
- INTRODUCCIÓN
- ALGORITMO DE ESCALADA
- ESCALADA ALEATORIA
- RECOCIDO SIMULADO
- ALGORITMOS GENÉTICOS

INTRODUCCIÓN



INTRODUCCIÓN

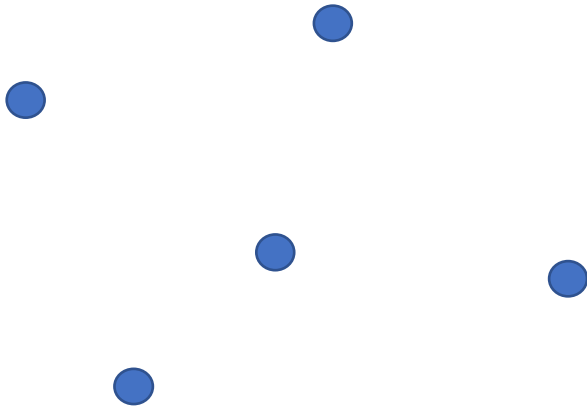
- Problema del viajante (TSP – travelling salesman problem)



- Estado: circuito (o tour) de las N ciudades, visitándolas una sola vez.
- Arcos simétricos
- Minimizar $f(e)$ = distancia recorrida

INTRODUCCIÓN

- Problema del viajante (TSP – travelling salesman problem)

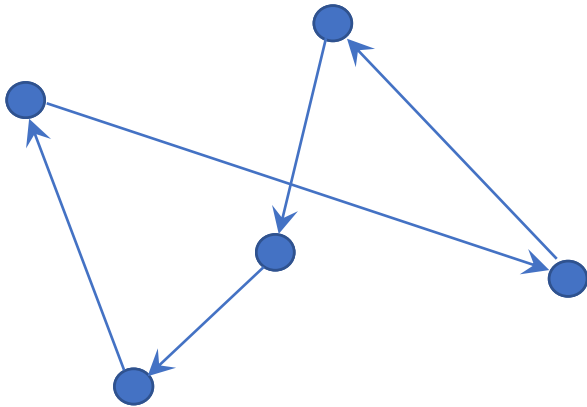


Vecindad 2-opt:

1. Eliminamos dos arcos no consecutivos $(x_1, x_2), (y_1, y_2)$.
2. Los sustituimos por $(x_1, y_1), (x_2, y_2)$.

INTRODUCCIÓN

- Problema del viajante (TSP – travelling salesman problem)

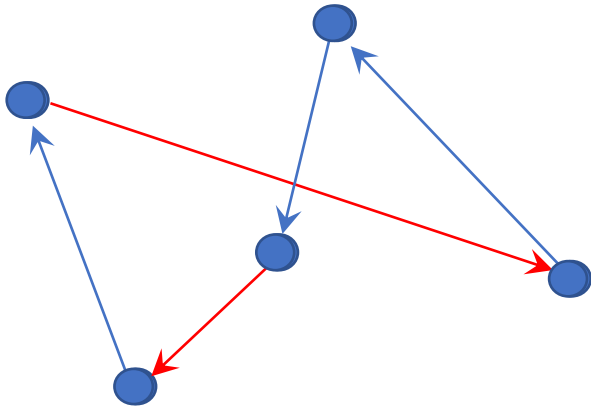


Vecindad 2-opt:

1. Eliminamos dos arcos no consecutivos $(x_1, x_2), (y_1, y_2)$.
2. Los sustituimos por $(x_1, y_1), (x_2, y_2)$.

INTRODUCCIÓN

- Problema del viajante (TSP – travelling salesman problem)

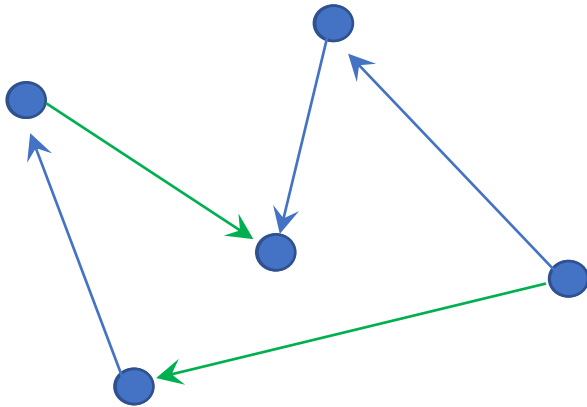


Vecindad 2-opt:

1. Eliminamos dos arcos no consecutivos $(x_1, x_2), (y_1, y_2)$.
2. Los sustituimos por $(x_1, y_1), (x_2, y_2)$.

INTRODUCCIÓN

- Problema del viajante (TSP – travelling salesman problem)

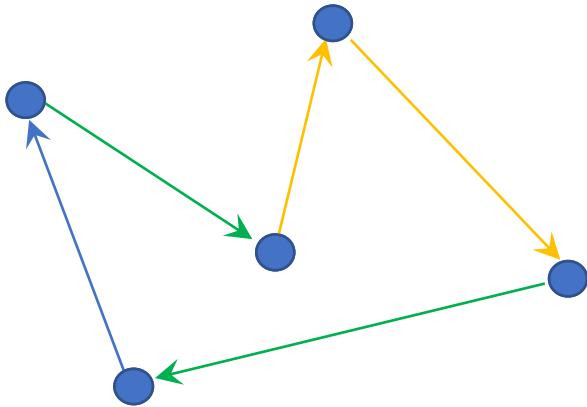


Vecindad 2-opt:

1. Eliminamos dos arcos no consecutivos $(x_1, x_2), (y_1, y_2)$.
2. Los sustituimos por $(x_1, y_1), (x_2, y_2)$.

INTRODUCCIÓN

- Problema del viajante (TSP – travelling salesman problem)

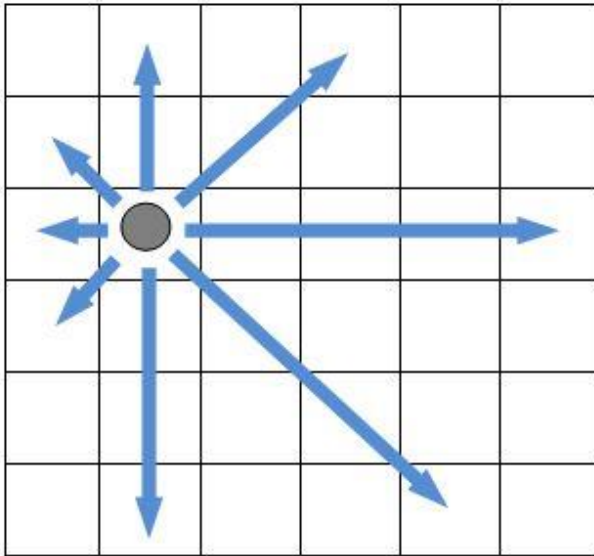


Vecindad 2-opt:

1. Eliminamos dos arcos no consecutivos $(x_1, x_2), (y_1, y_2)$.
2. Los sustituimos por $(x_1, y_1), (x_2, y_2)$.

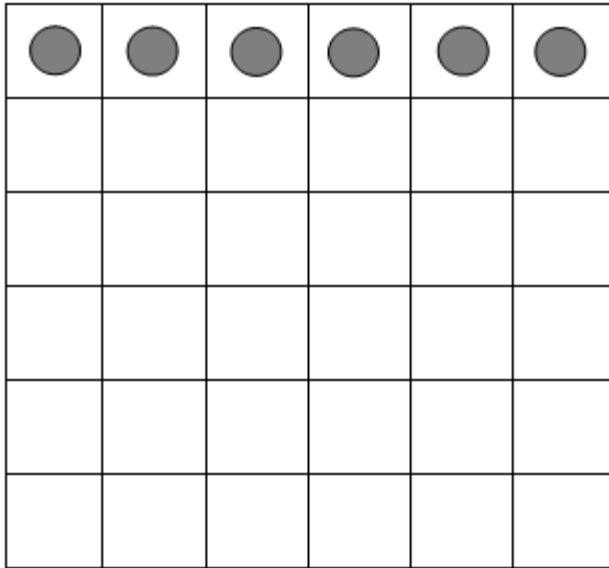
INTRODUCCIÓN

- Problema de las N reinas (satisfacción de restricciones)



INTRODUCCIÓN

- Problema de las N reinas (satisfacción de restricciones)

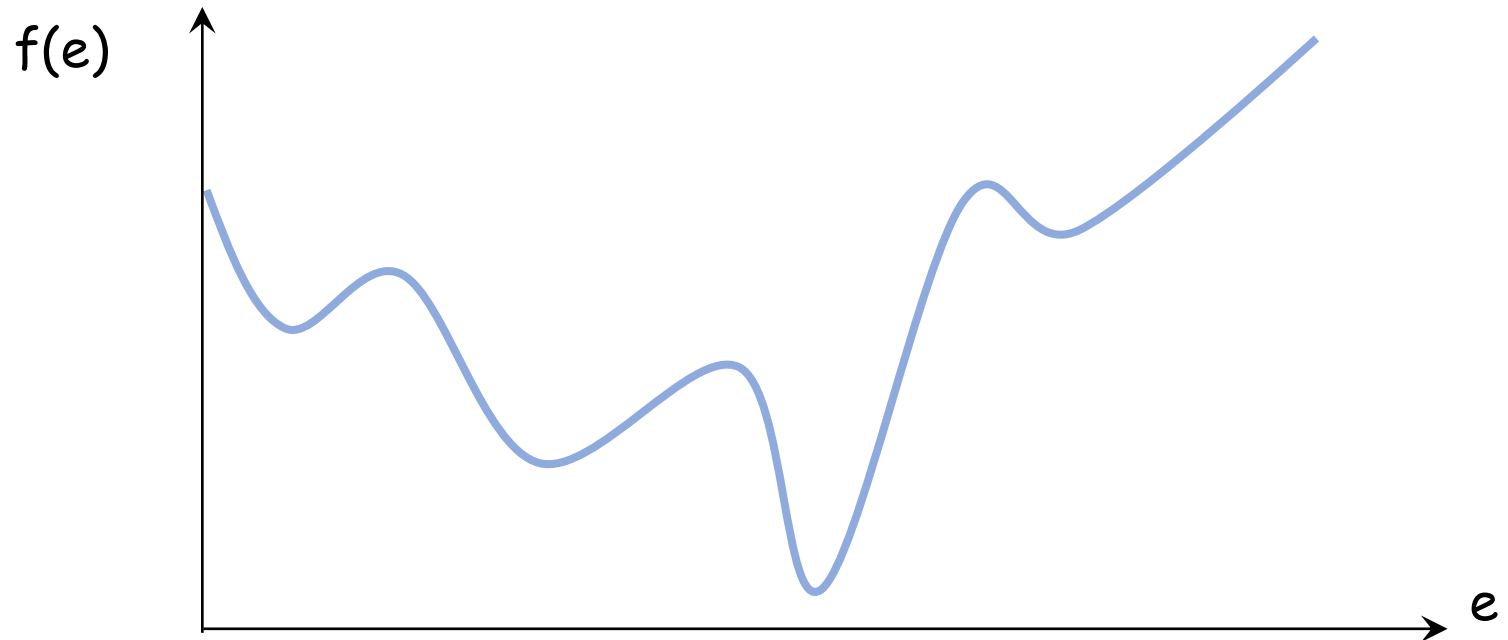


- Estado: una reina por columna
- Minimizar $f(e)$ = pares de reinas que incumplen una restricción

- INTRODUCCIÓN
- ALGORITMO DE ESCALADA
- ESCALADA ALEATORIA
- RECOCIDO SIMULADO
- ALGORITMOS GENÉTICOS

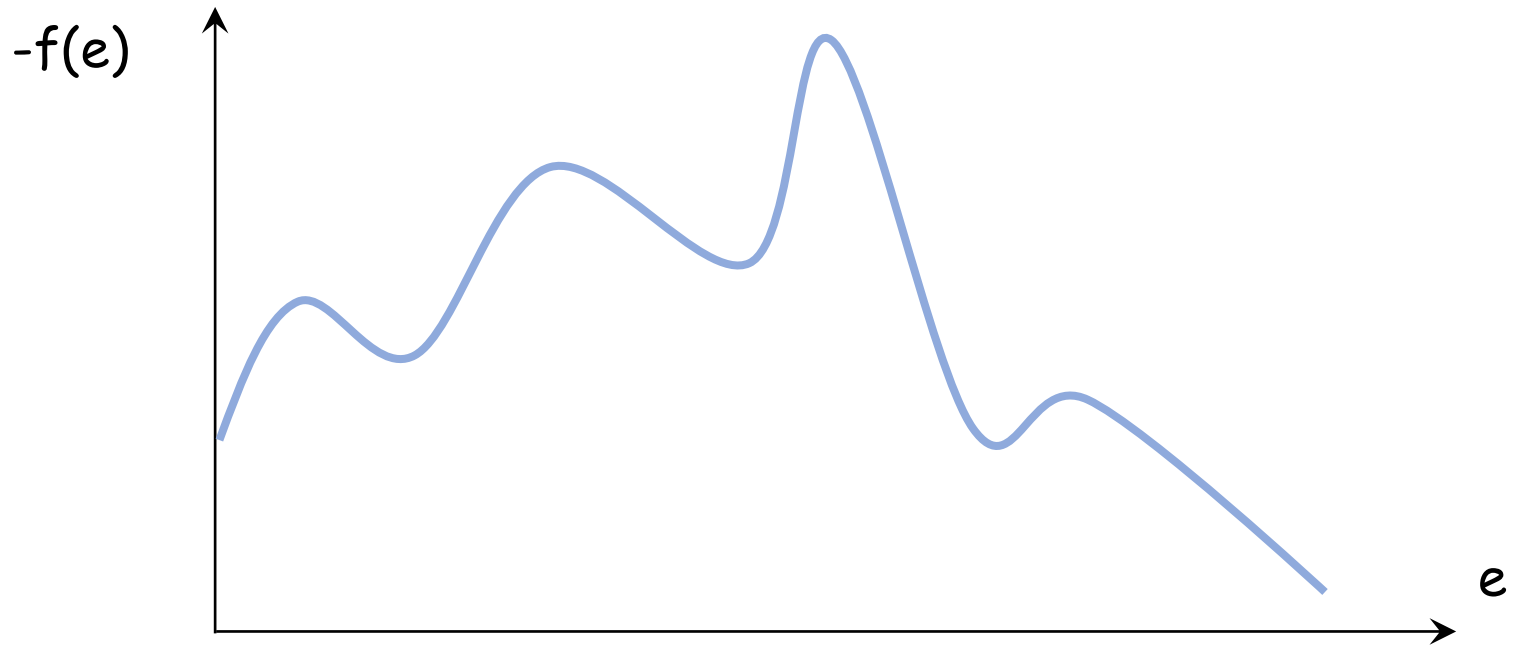
ALGORITMO DE ESCALADA

- Escalada (Hill climbing)



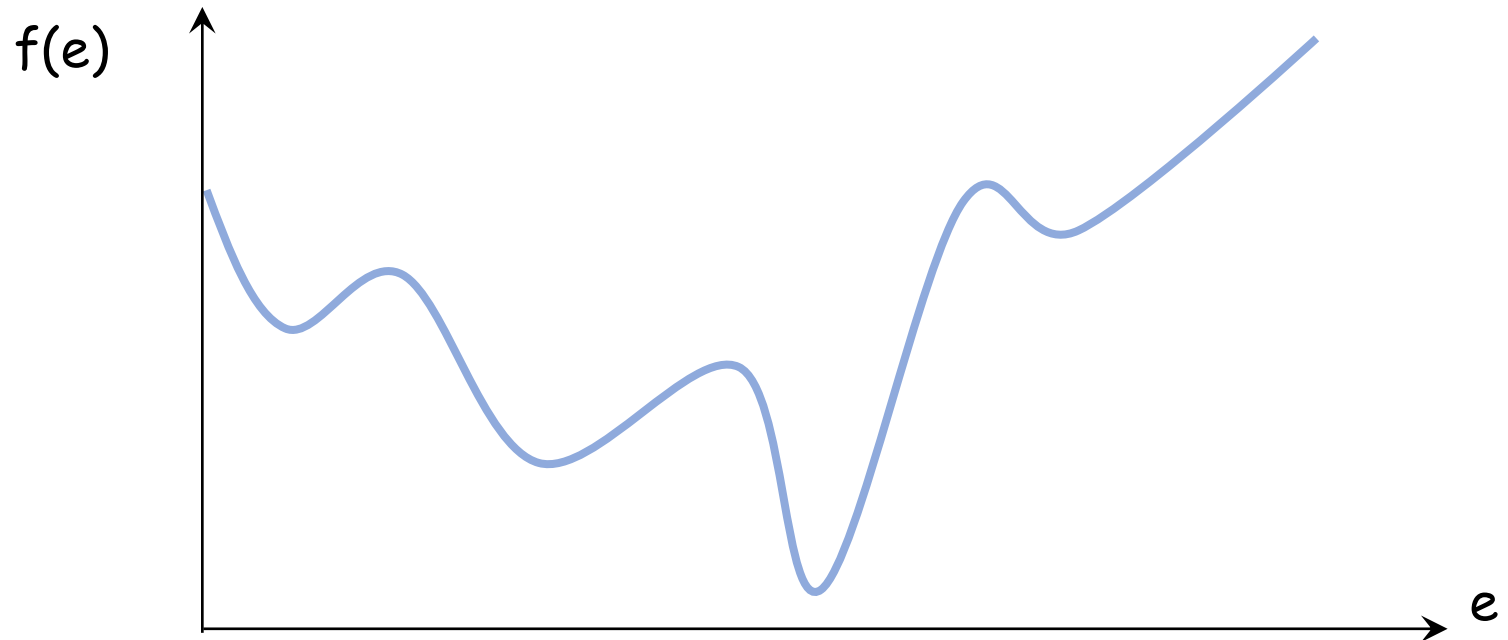
ALGORITMO DE ESCALADA

- Escalada (Hill climbing)



ALGORITMO DE ESCALADA

- Escalada (Hill climbing)



ALGORITMO DE ESCALADA

escalada (s) (versión para maximizar)

vecino \leftarrow mejor-sucesor(s);

Si $f(\text{vecino}) \leq f(s)$ | //se puede relajar temporalmente a <
| devolver(s)
sino
| escalada(vecino)

- INTRODUCCIÓN
- ALGORITMO DE ESCALADA
- ESCALADA ALEATORIA
- RECOCIDO SIMULADO
- ALGORITMOS GENÉTICOS

ESCALADA ALEATORIA

escalada-aleatoria (s) (versión para maximizar)

Mientras s no cambie o tiempo excedido

```
├   vecino ← sucesor-aleatorio(s)
├    $\Delta E \leftarrow f(\text{vecino}) - f(s)$ 
├   Si  $\Delta E > 0$ 
├   │   s ← vecino
```

Si tiempo excedido

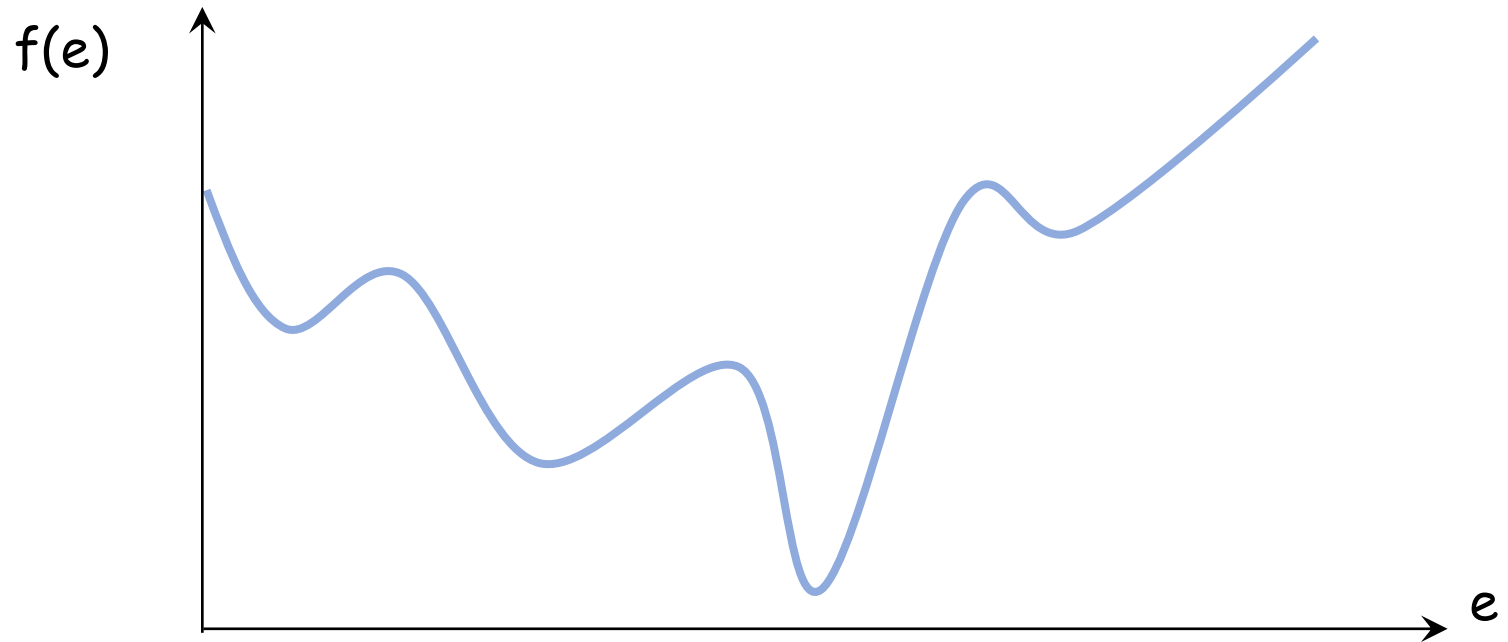
```
├   return(s)
```

escalada-aleatoria(s)

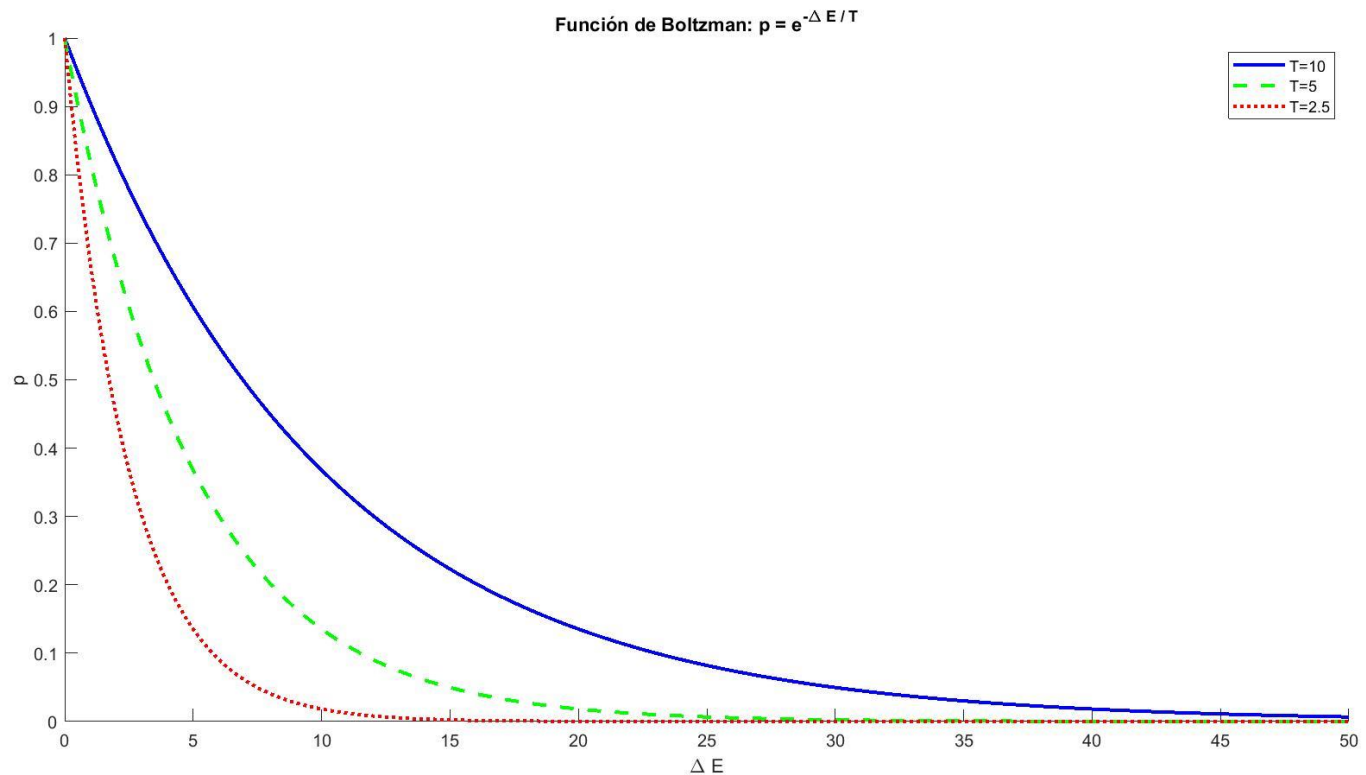
- INTRODUCCIÓN
- ALGORITMO DE ESCALADA
- ESCALADA ALEATORIA
- RECOCIDO SIMULADO
- ALGORITMOS GENÉTICOS

RECOCIDO SIMULADO

- Simulated annealing



RECOCIDO SIMULADO



Enfriado progresivo, p. ej. $T \leftarrow T \times 0.9$

RECOCIDO SIMULADO

recocido-simulado(s) (versión para maximizar)

$T \leftarrow T_0$

Durante un tiempo

 Durante un tiempo

 vecino \leftarrow vecino-aleatorio(s)

$\Delta E \leftarrow f(\text{vecino}) - f(s)$

 Si $\Delta E > 0$

$s \leftarrow \text{vecino}$

 sino

 asignar vecino a s con probabilidad $e^{\Delta E/T}$

 Reducir el valor de T

devolver(s)





EJEMPLOS

- Problema de las 4 reinas

●	●	●	●

EJEMPLOS

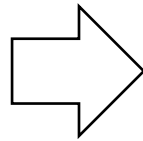
- Problema de las 4 reinas

			
2	1	1	2
2	2	2	2
2	3	3	2

EJEMPLOS

- Problema de las 4 reinas

●	●	●	●
2	1	1	2
2	2	2	2
2	3	3	2

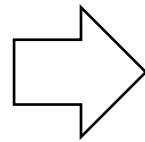


●	0	●	●
5	1	4	3
3	2	3	5
3	●	4	3

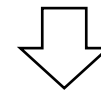
EJEMPLOS

- Problema de las 4 reinas

●	●	●	●
2	1	1	2
2	2	2	2
2	3	3	2



●	0	●	●
5	1	4	3
3	2	3	5
3	●	4	3

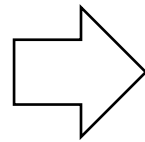


3	2	●	●
●	3	4	3
3	3	5	6
3	●	4	5

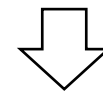
EJEMPLOS

- Problema de las 4 reinas

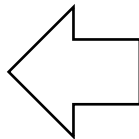
●	●	●	●
2	1	1	2
2	2	2	2
2	3	3	2



●	0	●	●
5	1	4	3
3	2	3	5
3	●	4	3



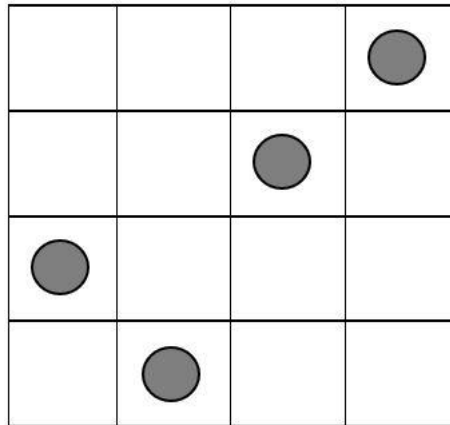
3	2	●	●
●	3	4	3
3	3	5	6
3	●	4	5



		●	
●			
			●
	●		





EJEMPLOS

- Problema de las 4 reinas (meseta)










EJEMPLOS

- Problema de las 4 reinas (meseta)

4	3	3	
4	3		3
	2	3	3
2		4	4

EJEMPLOS

- Problema de las 7 reinas (mínimo local estricto)

19	19	19	18	18	18	
18	19	19	18		18	18
19		19	19	17	17	18
16	18	18	19	18		17
	16	19	18	18	18	18
18	17		19	17	19	19
18	18	18		19	18	19

EJEMPLOS

- Recocido simulado

https://en.wikipedia.org/wiki/Simulated_annealing

<https://www.youtube.com/watch?v=SC5CX8drAtU>

- INTRODUCCIÓN
- ALGORITMO DE ESCALADA
- ESCALADA ALEATORIA
- RECOCIDO SIMULADO
- ALGORITMOS GENÉTICOS

ALGORITMOS GENÉTICOS

- Se guarda un conjunto de estados (población de individuos)
- Cada conjunto de estados se sustituye iterativamente por otro (generaciones)
- La generación de nuevos estados (individuos) se basa en la selección natural y la reproducción sexual

ALGORITMOS GENÉTICOS

- Representación: código genético (**conjunto de variables**)
- Se intenta **maximizar** una función de aptitud (fitness)
- Por ejemplo, para las N reinas:
 - Código genético: secuencia con la fila de cada reina
 - Función de aptitud: pares de reinas que satisfacen las restricciones.

ALGORITMOS GENÉTICOS

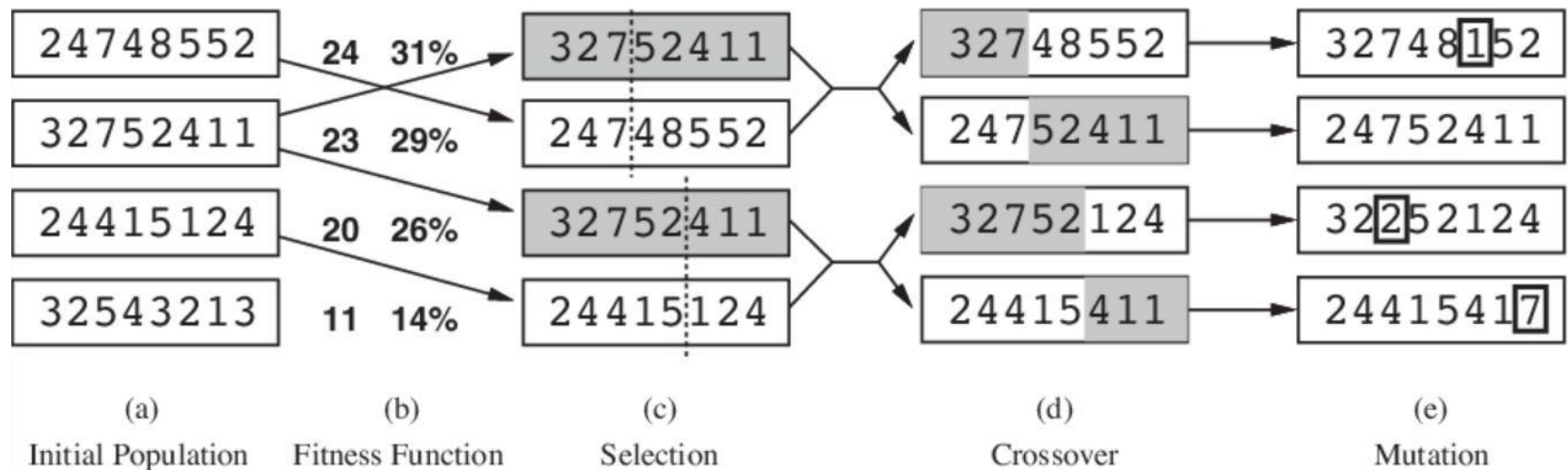


Imagen cortesía Russell & Norvig "Artificial intelligence: a modern approach"
 (<http://aima.cs.berkeley.edu/figures.html>)

ALGORITMOS GENÉTICOS

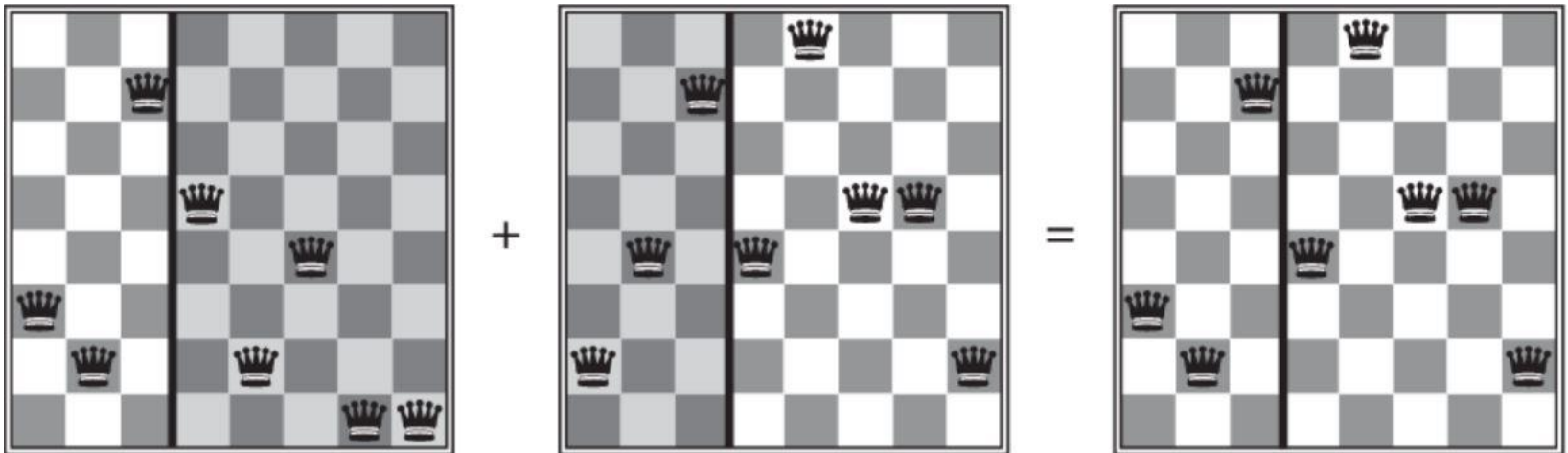


Imagen cortesía Russell & Norvig "Artificial intelligence: a modern approach"
(<http://aima.cs.berkeley.edu/figures.html>)

ALGORITMOS GENÉTICOS

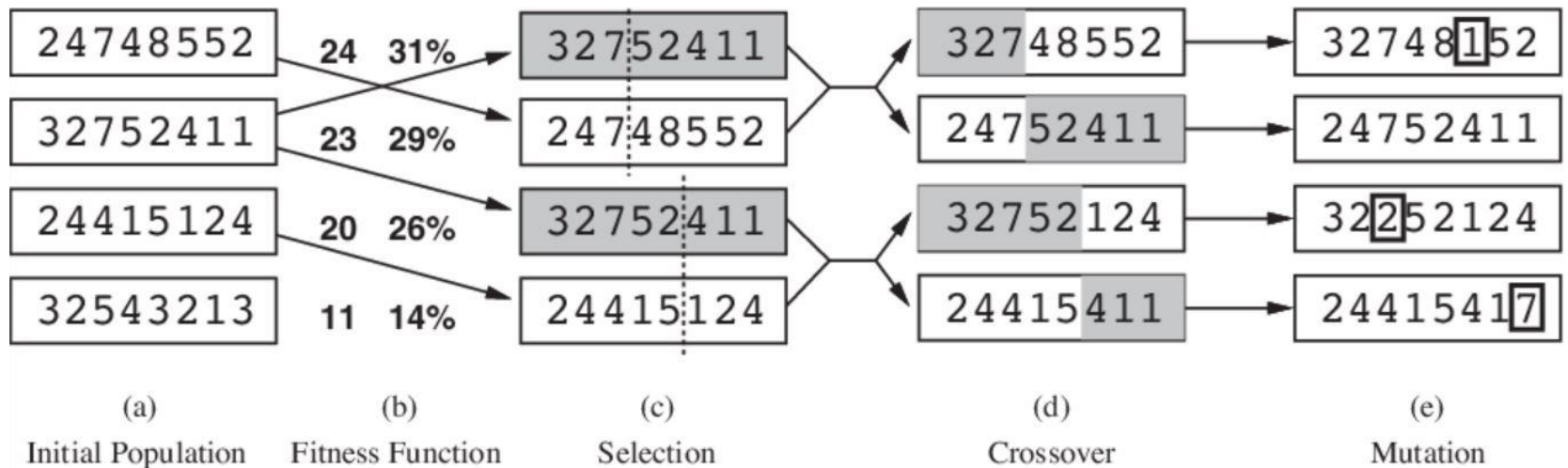


Imagen cortesía Russell & Norvig "Artificial intelligence: a modern approach"
(<http://aima.cs.berkeley.edu/figures.html>)

Búsqueda local

L. Mandow

Dpto. Lenguajes y Ciencias de la Computación

Universidad de Málaga

Abril, 2020