



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Búsqueda local

© Fernando Berzal, berzal@acm.org

Búsqueda local



- Ascensión de colinas
- Enfriamiento simulado
- Algoritmos genéticos



Ascensión de colinas



Ascensión de colinas simple

E: Estado activo

```
while (E no sea el objetivo
      y queden nodos por explorar a partir de E)
  Seleccionar operador R para aplicarlo a E
  Evaluar  $f(R(E))$ 
  if ( $f(R(E)) > f(E)$ )
    E = R(E)
```

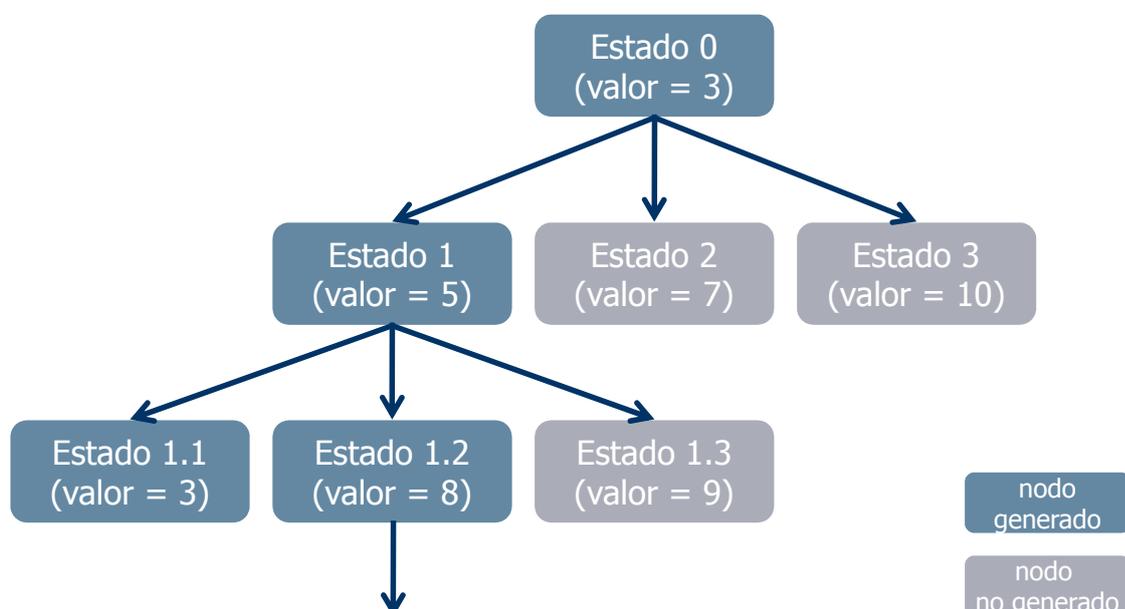


Ascensión de colinas



Ascensión de colinas simple

“Como subir al Everest con una niebla espesa y amnesia”



Ascensión de colinas



Ascensión de colinas por la máxima pendiente

E: Estado activo

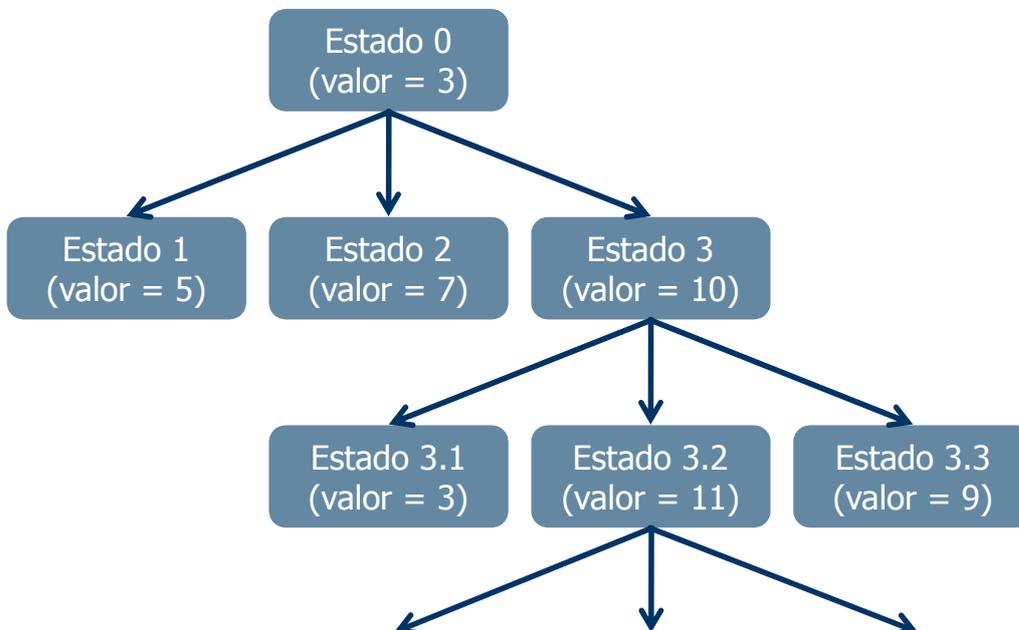
```
while (queden nodos por explorar a partir de E)
  Para todos los operadores  $R_i$ , obtener  $E_i=R_i(E)$ 
  Evaluar  $f(E_i)$  para todos los estados  $E_i=R_i(E)$ 
  Seleccionar  $E_{max}$  tal que  $f(E_{max}) = \max\{f(E_i)\}$ 
  if ( $f(E_{max})>f(E)$ )
    E =  $E_{max}$ 
  else
    return E
```



Ascensión de colinas



Ascensión de colinas por la máxima pendiente



Ascensión de colinas



Ascensión de colinas por la máxima pendiente

Observaciones

- Se realiza una búsqueda del mejor hijo del nodo actual E (para seleccionar el operador R más prometedor).
- Una vez elegido el operador R más prometedor, obtenemos el estado $E' = R(E)$ y se repite el proceso a partir del estado E'
- En memoria sólo tenemos que almacenar E , el hijo de E que estemos considerando en cada momento y el mejor hijo que hayamos encontrado.



Ascensión de colinas



Ascensión de colinas por la máxima pendiente

Observaciones

- El proceso se repite hasta que se encuentre una solución, hasta no podamos avanzar más o hasta que todos los hijos sean peores que el padre del que provienen.
- Este último criterio puede dar lugar a varios problemas debido a la existencia de máximos locales y mesetas en la función de evaluación heurística.

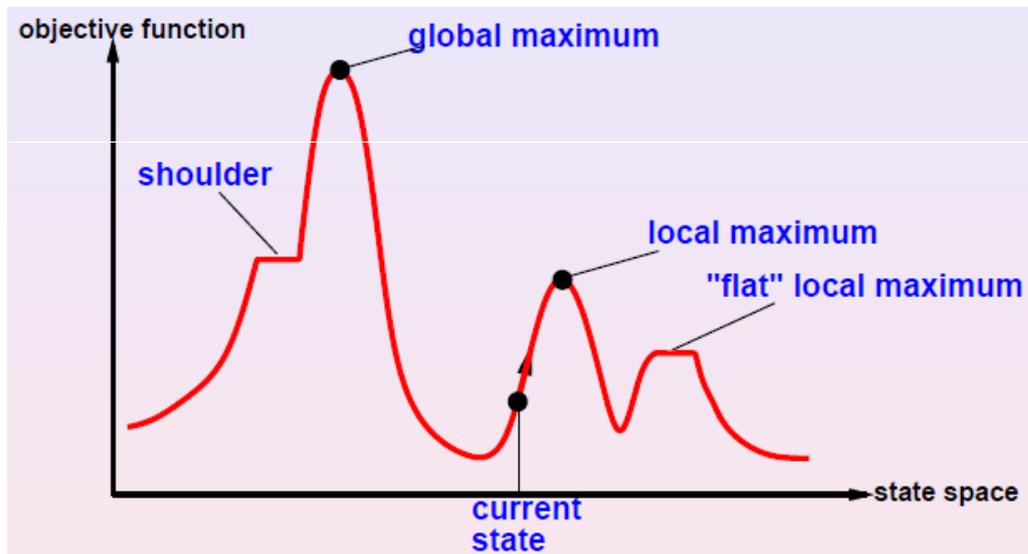


Ascensión de colinas



Ascensión de colinas por la máxima pendiente

Limitaciones



Ascensión de colinas



Ascensión de colinas por la máxima pendiente

Limitaciones

1	2	3
8		4
7	6	5

objetivo

3	2	1
8		4
7	6	5

$E_{\text{máximo local}}$

Máximo local

Todos los movimientos empeoran el valor de la función heurística.

1	2	3
6	7	4
	8	5

E_{meseta}

Meseta

Todos los movimientos dejan igual el valor de la función heurística.



Ascensión de colinas



Ascensión de colinas por la máxima pendiente Posibles soluciones

- Continuar la exploración de más niveles del árbol (no estaríamos ante una estrategia de búsqueda "local").
- Probar con distintos puntos de partida.
- "Dar saltos", aunque sólo sea de vez en cuando.

vg. enfriamiento simulado [simulated annealing]
algoritmos genéticos



Enfriamiento simulado



- Proceso de recocido del acero y cerámicas que consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas.
- El calor causa que los átomos aumenten su energía y que puedan así desplazarse de sus posiciones iniciales (un mínimo local de energía).
- El enfriamiento lento les da mayores probabilidades de recristalizar en configuraciones con menor energía que la inicial (mínimo global).



Enfriamiento simulado



IDEA

Escapar de los máximos local permitiendo movimientos malos. Gradualmente, tales movimientos decrecen en tamaño y frecuencia.

MECANISMO: Temperatura $T(i)$

- Empieza siendo alta y decrece hasta aproximarse a 0.
- A una temperatura $T(i)$, la probabilidad de cada estado se puede calcular usando la distribución de Boltzmann:

$$p(x) = \alpha e^{-\frac{E(x)}{kT}}$$



Enfriamiento simulado



Algoritmo

```
s ← s0; e ← E(s) // Estado y energía inicial
best ← s; ebest ← e // Mejor solución actual
k ← 0 // Número de iteración
while (k < kmax) and (e > emin)
  T ← temperature(k/kmax) // Temperatura actual
  snew ← neighbour(s) // Vecino aleatorio
  enew ← E(snew) // Energía asociada al vecino
  if P(e, enew, T) > random()
    s ← snew; e ← enew // Salto aleatorio al vecino
  if enew < ebest then
    sbest ← snew; ebest ← enew // Mejor solución encontrada
  k ← k + 1
return sbest // Resultado final
```



Algoritmos genéticos



- Se hace evolucionar una población de individuos (cada uno de los cuales representa una posible solución).
- La población se somete a acciones aleatorias semejantes a las de la evolución biológica (mutaciones y recombinaciones genéticas).
- Los individuos se seleccionan de acuerdo con una función de adaptación en función del cual se decide qué individuos sobreviven (los más adaptados) y cuáles son descartados (los menos aptos).



Algoritmos genéticos



Algoritmo

```
t ← 0
población(t) ← poblaciónInicial
EVALUAR(población(t))

while not (condición de terminación)
    t ← t + 1
    población(t) ← SELECCIONAR(población(t-1))
    población(t) ← CRUZAR(población(t))
    población(t) ← MUTAR(población(t))
    EVALUAR(población(t))

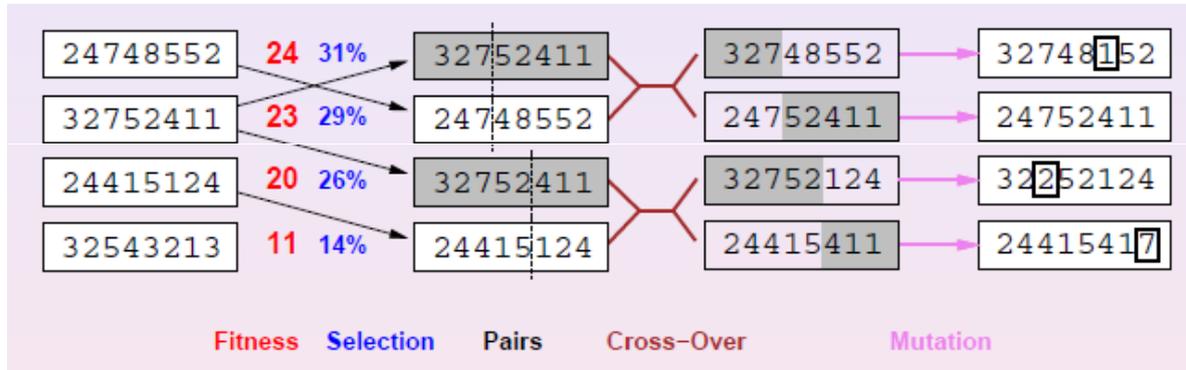
return población(t)
```



Algoritmos genéticos



Selección, cruce & mutación

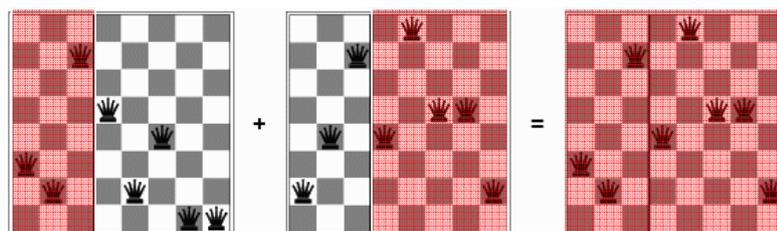


Algoritmos genéticos



Ejemplo: El problema de las N reinas

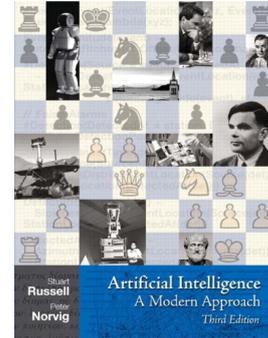
- Función de evaluación:
Número de parejas de reinas que no se atacan.
- Operador de cruce:



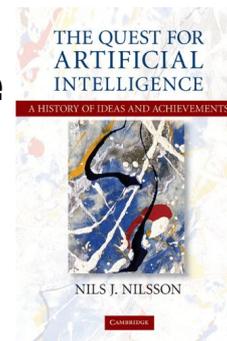
Bibliografía



- Stuart Russell & Peter Norvig:
**Artificial Intelligence:
A Modern Approach**
Prentice-Hall, 3rd edition, 2009
ISBN 0136042597



- Nils J. Nilsson
The Quest for Artificial Intelligence
Cambridge University Press, 2009
ISBN 0521122937



Bibliografía



Bibliografía complementaria

- Elaine Rich & Kevin Knight:
Artificial Intelligence.
McGraw-Hill, 1991.
- Patrick Henry Winston:
Artificial Intelligence.
Addison-Wesley, 1992.
- Nils J. Nilsson:
Principles of Artificial Intelligence.
Morgan Kaufmann, 1986.

