

# Búsqueda Local Estocástica

Tomás de la Rosa

## 1 Introducción

## 2 Algoritmos

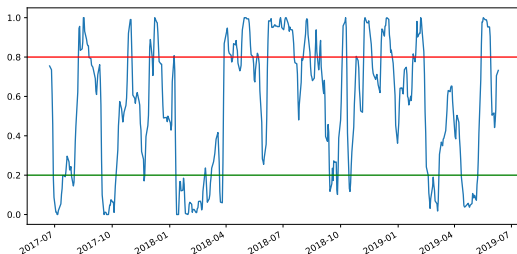
## 1 Introducción

## 2 Algoritmos

- **Búsqueda Local Estocástica:** Algoritmos de búsqueda en los que aleatorizar los procesos juega un papel fundamental en la capacidad de exploración
- Los procesos aleatorios pueden utilizarse por ejemplo:
  - ▶ En la construcción de una solución inicial que se mejorará posteriormente
  - ▶ Deshacer empates entre alternativas equivalentes
  - ▶ Selección probabilística entre un conjunto de candidatos grande

- La búsqueda local estocástica abarca un amplio espectro de técnicas de optimización
  - ▶ búsqueda local iterativa
  - ▶ simulated annealing
  - ▶ algoritmos genéticos

- Cuál es el conjunto de parámetros que históricamente maximiza el rendimiento utilizando la señal del oscilador estocástico
- Parámetros
  - ▶ ventana, días de suavizado
  - ▶ umbral de sobre-compra, sobre-venta



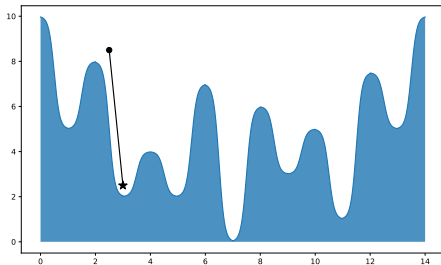
- Los algoritmos SLS parten de una solución inicial que se obtiene de forma *constructiva*
- Para la construcción se elige cada elemento de la solución de forma independiente a partir de una función de evaluación o heurística sencilla
- Ejemplo
  - ▶ TSP (*Travel Salesman Problem*): A cada paso elegir el siguiente punto más cercano que no haya sido elegido
  - ▶ Indicadores Análisis Técnico: Elegir aleatoriamente cada parámetro a partir de las alternativas disponibles

- Esquema general en el que se basan muchos algoritmos de SLS
- Necesita una función para evaluar la calidad de las soluciones ( $h(s)$ )
- En el contexto de búsqueda sistemática corresponde al algoritmo de *hill-climbing*
- Algoritmo:
  - ▶ El estado inicial es una solución válida
  - ▶ Para el estado actual:
    - 1 Generar los sucesores del estado actual y evaluarlos
    - 2 Reemplazar el estado actual por un sucesor que mejore evaluación
    - 3 Si no hay ningún sucesor mejor, parar.

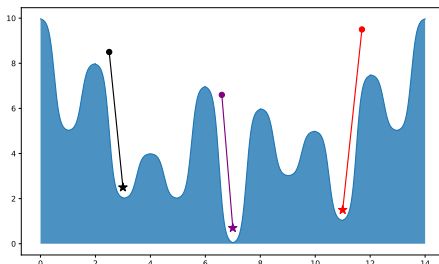


- Es la mejor solución en el “entorno” de la solución actual
- El algoritmo de mejora iterativa nos lleva al mínimo local de la solución inicial
- No hay garantías de optimalidad. Escapar de mínimos locales es la clave de la búsqueda local estocástica

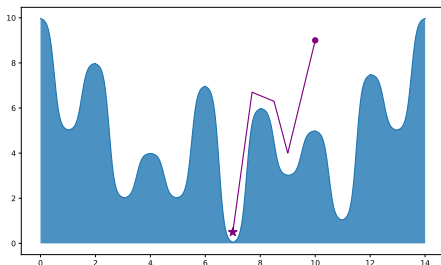
- Es la mejor solución en el “entorno” de la solución actual
- El algoritmo de mejora iterativa nos lleva al mínimo local de la solución inicial
- No hay garantías de optimalidad. Escapar de mínimos locales es la clave de la búsqueda local estocástica



- Una alternativa es re-iniciar la búsqueda cada vez en un entorno diferente para así llegar a mínimos locales en todo el espacio de búsqueda
- Necesitamos un generador aleatorio de soluciones iniciales, que mejoraremos con el procedimiento de mejora iterativa



- Otra alternativa es permitir que la mejora iterativa pueda empeorar la calidad de la solución actual para que se desvíe hacia otras zonas del espacio de búsqueda
- Con cierta probabilidad dejamos que el proceso elija un camino aleatorio



1 Introducción

2 Algoritmos

- De forma ocasional permite explorar aleatoriamente regiones que no seguirían el sentido de la función de evaluación
- Esquema:
  - 1 Con probabilidad  $w$  se realiza un paso o camino aleatorio
  - 2 Con probabilidad  $1 - w$  se realiza un paso de mejora iterativa
- Fácil de implementar y depende de un único parámetro  $w$

- Inspirado en el proceso de recocido de metales o vidrio, donde una temperatura alta va decreciendo lentamente
- En el algoritmo la temperatura  $T$ , representa la probabilidad de empeorar la solución actual desviándose de la mejora iterativa
- Después de llegar a una temperatura muy baja, se comporta como una mejora iterativa hasta que encuentra el mínimo local



## ● Algoritmo

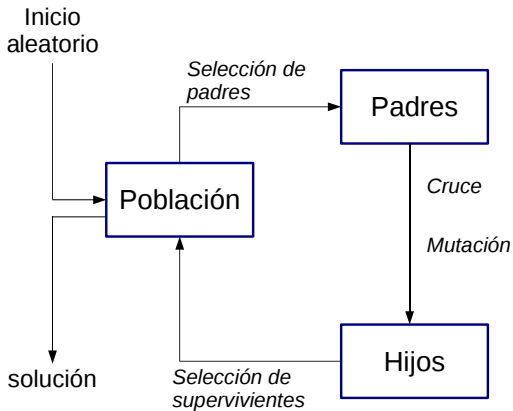
- ▶ Partimos de una solución inicial (subóptima)
- ▶ Necesitamos una función de evaluación  $f(s)$  y la generación de sucesores

```
s = solucion inicial
for i=1 to N
    T = i/N
    cand = random_choice(sucesores(s))
     $\Delta E = f(s) - f(cand)$ 
     $c = e^{\Delta E / T}$ 
    If  $\Delta E > 0$  or  $c > \text{random}(0,1)$  then
        s = cand
return s
```



- Realiza un proceso de mejora iterativa con modificaciones
- Cuando no hay sucesor que mejore la función de evaluación se puede empeorar la solución actual
- Se guarda en memoria una lista (tabú) con los últimos cambios a la solución que han sido explorados
- En la selección de sucesores se excluyen los que estén en la lista tabú.
- La lista tabú puede limitarse:
  - ▶ memoria a corto plazo: se está en la lista un número de iteraciones
  - ▶ cambio de contexto: se re-inicia cuando se mejora la solución actual

- Utilizar simultáneamente varias soluciones como estado actual, para aumentar la diversidad de la búsqueda
- Se aplica la lógica de un algoritmo básico (ej. mejora iterativa aleatorizada) a cada una de las soluciones
- El beneficio de la paralelización no tiene que por qué ser mejor que ejecutar en paralelo varios algoritmos individualmente
- **Algoritmos Genéticos:** Algoritmos de búsqueda local inspirados en el proceso de selección natural



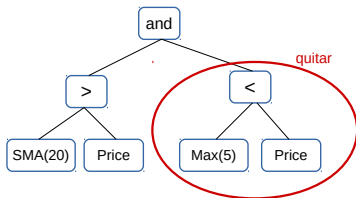
INSTITUTO  
**BME** X



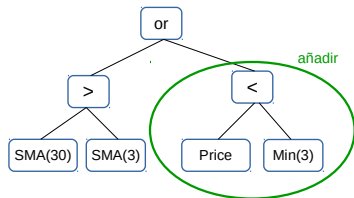
- Se encuentra una solución con un valor de la función de adecuación por encima de un umbral
- Se realiza un número determinado de iteraciones (generaciones)
- Se alcanza un límite de tiempo de cómputo
- El valor de la función de adecuación no cambia significativamente de una generación a otra
- Por decisión manual
- Combinación de varias

- Algoritmos Genéticos: cadenas de bits
- Estrategias Evolutivas: números reales
- Programación Evolutiva: Máquinas de Estados Finitos
- Programación genética: árboles

Padre 1



Padre 2



Hijo

