

Inteligencia Artificial

Técnicas de Búsqueda Incompleta: Tabu Search y Simulated Annealing

Elizabeth Montero Ureta

Departamento de Informática
Universidad Técnica Federico Santa María

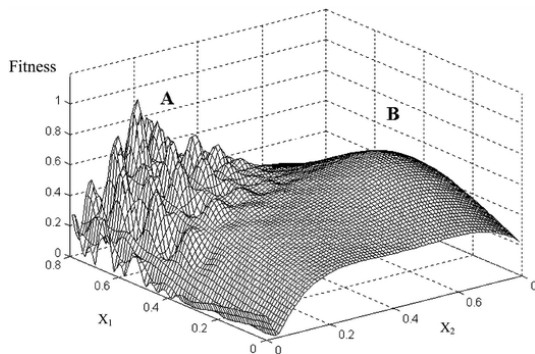
2do Semestre 2018

- 1 Escape de óptimos locales
- 2 Búsqueda Tabú (Tabu Search)
- 3 Simulated Annealing (Recocido Simulado)
- 4 Heurísticas K-opt

Escape de óptimos locales

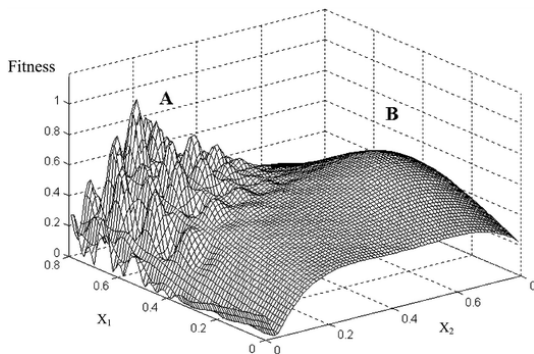
Problema

- Estancamiento de la búsqueda en óptimos locales



Problema

- Estancamiento de la búsqueda en óptimos locales



Solución

- Opción 1: Reiniciar la búsqueda → pérdida de información
- Opción 2: Aceptar movimientos que empeoran la calidad de la solución actual
 - ▶ Pueden producir ciclos en la búsqueda.

Búsqueda Tabú (Tabu Search)

- Búsqueda Tabú (Glover, 86) es una estrategia para resolver problemas de optimización combinatoria. Algo muy parecido sugirió Hansen al mismo tiempo (*steepest ascent/mildest descent*).
- Utiliza una búsqueda local con memoria a corto plazo
 - ▶ El proceso es capaz de escapar de óptimos locales y evitar ciclos utilizando la información almacenada en la memoria.
- La memoria de corto plazo esta representada por la lista tabú, la cual registra las últimas soluciones visitadas e impide volver a ellas en los próximos movimientos.

- Por razones de eficiencia, no se guarda la solución completa en la lista tabú, sino una parte de sus atributos. Se define una lista tabú de condiciones.
- La lista tabú se actualiza normalmente en forma FIFO.
- El largo de la lista tabú controla la memoria del proceso de búsqueda.
 - ▶ Una lista tabú corta, controla áreas reducidas del espacio de búsqueda y una larga, fuerza a una búsqueda en áreas mayores.

- Por razones de eficiencia, no se guarda la solución completa en la lista tabú, sino una parte de sus atributos. Se define una lista tabú de condiciones.
- La lista tabú se actualiza normalmente en forma FIFO.
- El largo de la lista tabú controla la memoria del proceso de búsqueda.
 - ▶ Una lista tabú corta, controla áreas reducidas del espacio de búsqueda y una larga, fuerza a una búsqueda en áreas mayores.
- El largo de la lista puede cambiar a lo largo del proceso de búsqueda.

Procedure tabu search

initialize s_c at random

initialize $tabu_{list}$ as empty

initialize $s_{best} \leftarrow s_c$

Repeat

$s_v \leftarrow$ select from $\mathcal{N}(s_c)$ the best non tabu point

$s_c \leftarrow s_v$

update $tabu_{list}$

If s_c is better than s_{best} **then**

$s_{best} \leftarrow s_c$

Until $terminationCriterion()$

EndProcedure

- Considerando el problema de la mochila

$$\begin{aligned} \text{máx } & 18 \cdot x_1 + 25 \cdot x_2 + 11 \cdot x_3 + 14 \cdot x_4 \\ \text{s.a. } & 2 \cdot x_1 + 2 \cdot x_2 + x_3 + x_4 \leq 3 \\ & x_1, x_2, x_3, x_4 \in \{0, 1\} \end{aligned}$$

- Proponga una representación acorde al problema
- Proponga un movimiento acorde al problema
- Describa el proceso de búsqueda realizado por Búsqueda Tabú.
 - ▶ Criterio de parada: 5 iteraciones
 - ▶ Largo de la lista: 2

- Representación binaria de largo n , donde n corresponde a la cantidad de objetos.

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
|---|---|---|---|

- Movimiento: Cambio de una variable desde 0 a 1, ó desde 1 a 0
- Observación: Se trabaja sólo con soluciones factibles

- Intensificación: Búsqueda local.
- Diversificación: Aceptando soluciones de peor calidad.
- Balance estático a través del tamaño de la lista tabú.
 - ▶ Lista tabú larga: fomenta la diversificación a zonas más alejadas del espacio de búsqueda.

- Intensificación: Búsqueda local.
- Diversificación: Aceptando soluciones de peor calidad.
- Balance estático a través del tamaño de la lista tabú.
 - ▶ Lista tabú larga: fomenta la diversificación a zonas más alejadas del espacio de búsqueda.
- Cambios dinámicos en el largo de la lista tabú (Reactive Tabu Search).
 - ▶ El largo de la lista varía según las propiedades de la trayectoria en el espacio de búsqueda.

- En el proceso se pierde información y buenas soluciones pueden ser excluidas del conjunto permitido.
- Para reducir este problema, se define un *criterio de aspiración* que permitiría a una solución estar dentro del conjunto de soluciones permitidas aún cuando figure en la lista tabú.

Procedure tabu search

initialize s_c at random

initialize $tabu_{list}$ as empty

initialize $s_{best} \leftarrow s_c$

Repeat

$s_v \leftarrow$ select from $\mathcal{N}(s_c)$ the best point that
satisfies the aspiration criteria or the best non tabu point

$s_c \leftarrow s_v$

update $tabu_{list}$

If s_c is better than s_{best} then

$s_{best} \leftarrow s_c$

Until $terminationCriterion()$

EndProcedure

Simulated Annealing (Recocido Simulado)

- Se basa en el trabajo de Metropolis et al. (1953) en el campo de la termodinámica estadística.
- Modelamiento del proceso de recocido simulando los cambios energéticos en un sistema de partículas conforme decrece la temperatura, hasta que converge a un estado estable (congelado).

- Permitir movimientos a soluciones que empeoren la función de evaluación de forma de poder escapar de los óptimos locales.
- Permitir movimientos que empeoran la calidad de la función de evaluación de acuerdo a una probabilidad asociada a la temperatura del sistema.
- Al comienzo, la temperatura es alta y cualquier transición entre estados es permitida y soluciones que empeoren la función de evaluación pueden ser aceptadas con mayor probabilidad que más tarde cuando la temperatura disminuye.
- Mientras que las soluciones que mejoran la función de evaluación siempre son aceptadas.

Procedure simulated annealing $t \leftarrow 0$ initialize T initialize s_c at randominitialize $s_{best} \leftarrow s_c$ **Repeat****Repeat** $s_n \leftarrow$ select a new point in $\mathcal{N}(s_c)$ **If** $f(s_n)$ is better than $f(s_c)$ **then** $s_c \leftarrow s_n$ **Else If** $random([0, 1]) < e^{\frac{\Delta_{eval}}{T}}$ **then** $s_c \leftarrow s_n$ **If** s_c is better than s_{best} **then** $s_{best} \leftarrow s_c$ **Until** *haltingCriterion()* $T \leftarrow g(T, t)$ $t \leftarrow t + 1$ **Until** *terminationCriterion()***EndProcedure**

$$P(\text{temperatura, solución nueva, solución actual}) = P(.)$$

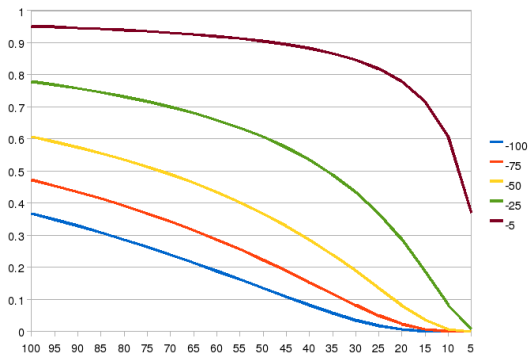
$$P(.) = e^{(\Delta_{eval} / \text{temperatura})} \quad (1)$$

donde

Δ_{eval} = Se considera siempre como un valor negativo, pues implica un empeoramiento de la calidad de las soluciones.

Acerca de la Temperatura:

- Temperatura versus Probabilidad de aceptación
- Cada línea muestra un valor distinto de Δ_{eval}



- La temperatura del sistema es controlada con enfriamientos sucesivos (función logarítmica) y recalentamientos periódicos, que permiten escapar de óptimos locales.
- Enfriamiento: Cada cierto número de iteraciones

$$T_{i+1} = \alpha \cdot T_i \quad (2)$$

donde $0 < \alpha < 1$, se recomienda utilizar $\alpha \in \{0.8; 0.99\}$

- Intensificación: Búsqueda local.
- Balance dinámico controlado a través del valor de la temperatura: Inicialmente alta diversificación y al final del algoritmo baja diversificación.
 - ▶ Alta Temperatura: gran diversificación
 - ▶ Baja Temperatura: poca diversificación.

- Intensificación: Búsqueda local.
- Balance dinámico controlado a través del valor de la temperatura: Inicialmente alta diversificación y al final del algoritmo baja diversificación.
 - ▶ Alta Temperatura: gran diversificación
 - ▶ Baja Temperatura: poca diversificación.
- Diversificación: Técnicas de recalentado y posterior enfriamiento (enfriamiento no monótono).

- Considerando el problema de la mochila

$$\begin{aligned} \text{máx } & 18 \cdot x_1 + 25 \cdot x_2 + 11 \cdot x_3 + 14 \cdot x_4 \\ \text{s.a. } & 2 \cdot x_1 + 2 \cdot x_2 + x_3 + x_4 \leq 3 \\ & x_1, x_2, x_3, x_4 \in \{0, 1\} \end{aligned}$$

- Proponga una representación acorde al problema
- Proponga un movimiento acorde al problema
- Describa el proceso de búsqueda realizado por Simulated Annealing.
 - ▶ Criterio de parada: 4 iteraciones
 - ▶ Temperatura Inicial: $t=10$ (fija durante las 4 iteraciones)

- Representación binaria de largo n , donde n corresponde a la cantidad de objetos.

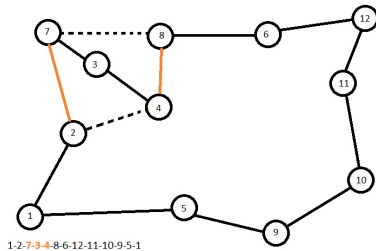
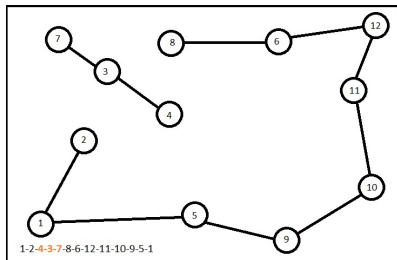
| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
|---|---|---|---|

- Movimiento: Cambio de una variable desde 0 a 1, ó desde 1 a 0
- Observación: Se trabaja sólo con soluciones factibles

Heurísticas K-opt

- Definición K-opt (K-exchange): Es un procedimiento que reemplaza K arcos de un tour de TSP por K nuevos arcos, tal que el tour resultante es un tour factible del TSP.

2-opt



3-opt

