# **1 Enfriamiento Simulado**

## Fundamentos

El enfriamiento simulado es un algoritmo de búsqueda por entornos con un criterio probabilístico de aceptación de soluciones basado en Termodinámica.

* Un modo de evitar que la búsqueda local finalice en óptimos locales, es permitir que algunos movimientos sean hacia soluciones peores
* Si la búsqueda está avanzando realmente hacia una buena solución, deben controlarse estos movimientos de “escape de óptimos locales”
* En el caso de Enfriamiento simulado ES, se realiza controlado la frecuencia de estos mediante una función de probabilidad que hará disminuir la probabilidad de moverse hacia soluciones peores conforme avanza la búsqueda
* Se aplica la filosofía habitual de búsqueda de diversificar al principio, e intensificar al final.

Es una técnica muy fácil pero puede quedar atrapada en óptimos locales, ocurre sobre todo si la temperatura es muy baja con respecto al “salto”

## Algoritmo de Metrópolis

* El fundamento de este control se basa en el trabajo de Metrópolis en el campo de la termodinámica estadística.
* Metrópolis modeló el proceso de enfriamiento simulando los cambios energéticos de un sistema de partículas conforme decrece la temperatura hasta que converge a un estado estable. Las leyes de la termodinámica dicen que a una temperatura t, al probabilidad de un incremento enérgico de magnitud δE se puede aproximar por:
  + P[δE] = exp(- δE/kt)
  + Siendo k una constante física denominada Boltzmann
* En este modelo, se genera una perturbación aleatoria en el sistema y se calculan los cambios de energía resultantes: si hay una caída energética, el cambio se acepta automáticamente, por el contrario si se produce un incremento, el cambio será aceptado con una probabilidad que se indica en la expresión anterior.

## 1.3 Algoritmo Básico de ES

* En cada iteración se genera un número concreto de vecinos
* Cada vez que se genera un vecino se aplica un criterio de aceptación para ver si sustituye a la solución actual
* Si la solución es mejor se acepta, como en una búsqueda local clásica
* Si es peor, puede que lo sustituya o no, depende de la probabilidad de aceptación, que está basada en la diferencia de los costes entre las soluciones y la temperatura:
  + Paceptación= exp(-δ/T)
* A mayor temperatura, más probabilidad de aceptar soluciones peores. Así al pin ripio del algoritmo se aceptan más que al final
* A menor diferencia de costes, mayor probabilidad de aceptación.
* Una vez finalizada la iteración, es decir al generar los vecinos se enfría la temperatura y se pasa a la siguiente iteración.

## Parámetros y componentes

Existen varias formas de representar un problema como

* Vector ordenado de números enteros (viajante de comercio)
* Vector binario(Mochila)
* Vector de números reales.

Así como mecanismos para generación de la solución inicial:

* Uso de técnicas eficiente para obtenerla
* Uso de conocimiento experto
  + Ej: Greedy.

También se dispone de mecanismos de transición de soluciones:

* Generación de una nueva solución
  + Definición del conjunto de vecinos
  + Selección de un elemento de dicho conjunto
* Calculo de la diferencia de costos entre las soluciones actual y la vecina
* Aplicación del criterio de aceptación

Secuencia de enfriamiento:

1. Valor inicial del parámetro de control (temperatura)
   1. No debería ser valores fijos independientes al problema
2. Mecanismos de enfriamiento. Existen varios como:
   1. Enfriamiento basado en sucesivas temperaturas descendentes prefijadas
   2. Enfriamiento con descenso constante de temperaturita
   3. Descenso geométrico
   4. Criterios e Boltzmann o Cauchy
3. Velocidad de enfriamiento
   1. Debe ser suficientemente grande como para que el sistema llegue a alcanzar su estado estacionario para esa temperatura
   2. Lo habitual es que sea un valor fijo, pero en algunas variantes se permite decidir cuándo es mejor finalizar la iteración actual y enfriar
4. Condición de parada
   1. En teoría el algoritmo debería finalizar cuando T=0. En la práctica se para :
      1. Cuando T alcanza o está debajo de un valor final T
      2. Después de un número fijo de iteraciones
   2. Como es difícil dar valor final de T se suele usar número fijo de iteraciones
   3. Otra opción es parar cuando no se haya aceptado ningún vecino de los generados en la iteración actual.

# **Algoritmo de búsqueda Tabú**

## 2.1 Introducción

La búsqueda tabú es un procedimiento de búsqueda por entornos cuya característica distintiva es el uso de memoria adaptativa y estrategias especiales de resolución de problemas.

La memoria adaptativa permite:

* Restringir el entorno de búsqueda
* Introducir mecanismos de reinicialización de la búsqueda mediante intensificación sobre zonas del espacio de búsqueda ya visitadas, o diversificación sobre zonas del espacio poco visitadas.

## 2.2 Estructura de la búsqueda tabú

**Fundamentos**

La búsqueda tabú TS es una técnica de búsqueda por entornos caracterizada por dos aspectos principales:

* Permite movimientos de empeoramiento para escapar de óptimos locales, para evitar ciclos incorpora un mecanismo de generación de vecinos modificado que evita la exploración de zonas del espacio que ya han sido visitadas. Entorno Tabú Restringidos
* Emplea mecanismos de reinicialización para mejorar la capacidad del algoritmo para la exploración- explotación del espacio de búsqueda.
* Para realizar estas dos tareas, hace uso de dos estructuras de memoria adaptativas distintas:
  + Memoria a corto plazo o Lista Tabú: Guarda información que permite guiar la búsqueda de forma inmediata, generando entornos restringidos
  + Memoria de largo plazo: Guarda información que permite guiar la búsqueda a posteriori, después de una primera etapa en la que se han realizado una o varias ejecuciones del algoritmo usando memoria a corto plazo
    - La información guardada en esta memoria se usa para comenzar con la búsqueda desde ora solución inicial de acuerdo a dos filosofías distintas
      * Intensificar la búsqueda, volviendo a visitar zonas del espacio prometedoras (con buenas soluciones) exploradas parcialmente
      * Diversificar la búsqueda, visitando nuevas zonas no explotadas aun.

**Memoria de corto plazo: Generación de entornos restringidos en la búsqueda tabú**

* La TS extienda la búsqueda local, en cada iteración se acepta siempre el mejor vecino sea mejor o peor
* La memoria a corto plazo permite a la TS determinar la lista de entorno disponible y así organizar como explora el espacio.
* Las soluciones admitidas en esta lista de entorno disponible dependen de la estructura de la lista tabú:
  + Lista de soluciones tabú: Se identifican soluciones ya visitadas y se marcan como tabú, eliminándolas del vecindario
  + Lista de movimientos tabú: Se eliminan del entorno todos los vecinos resultantes de aplicar sobre la solución actual un movimiento realizado anteriormente.
  + Lista de valores de atributos tabú: Se eliminan del entorno todos aquellos vecinos con un par (atributo, valor) determinado que ya presentara alguna solución explorada anteriormente

Los atributos o movimientos seleccionados son designados como tabú-activos y las soluciones que los contienen se convierten en soluciones tabú.

Tenencia Tabú:

* Un atributo/movimiento o solución que se haya incluido en la lista tabú en algún momento de la búsqueda no permanece para siempre
* La tenencia tabú es el intervalo de tiempo durante el que un atributo/movimiento permanece tabú-activo o una solución es tabú
* Este parámetro se mide en número de iteraciones.

Niveles de aspiración:

* El criterio de aspiración introduce un elemento de flexibilidad en la búsqueda tabú.
* El estado tabú de un movimiento/atributo puede ser ignorado si se cumplen ciertas condiciones en forma de niveles de aspiración. Una solución tabú dejará de serlo si supera cierto nivel de aspiración.

Estrategias para la lista de candidatos:

* Las estrategias se usan para restringir el número de vecinos examinados en una iteración dada, esto se hace cuando la lista es grande o la evaluación de elementos costosa
* Se busca el mejor movimiento disponible que pueda ser determinado con una cantidad apropiada de esfuerzo
* De esta forma no se genera el entorno reducido completo, sino una parte y se toma el mejor vecino.

Estructuras de memoria de corto plazo:

* Para facilitar la gestión de la tenencia tabú de los atributos lo mejor es implementar una lista circular
* Cuando se llena la lista, se comienza a insertar de nuevo por el principio borrando lo que hubiera en esas posiciones. De esta forma la tenencia es igual al tamaño de la lista.

**Memoria de largo plazo: Intensificación y Diversificación de la Búsqueda**

* En algunas aplicaciones las componentes de memoria TS de corto plazo son suficiente para producir soluciones de muy alta calidad
* No obstante la TS se vuelve más potente incluyendo memoria a largo plazo, y sus estrategias de reinicializacion asociadas
* La memoria de largo plazo se puede usar de dos modos:
  + Estrategia de intensificación. Se basan en una reinicializacion de la búsqueda que efectúa un regreso a regiones atractivas del espacio de para explotarlas. Se mantiene un registro de las mejores soluciones visitadas, se actualiza cuando se encuentra una mejor que la global
  + Estrategia de diversificación. Conducen la búsqueda hacia nuevas regiones del espacio de búsqueda no exploradas, la búsqueda se reinicia cuando se estanca, partiendo de una solución no visitada. Normalmente la solución inicial se genera a partir de la memoria de una memoria de frecuencias.
* Una estructura muy empleada es la memoria de frecuencias, que registra el número de veces que cada valor de un atributo ha pertenecido a soluciones visitadas en la búsqueda.

**Estructuras de memoria de largo plazo (frecuencias)**

* Si las variables son binarias se puede usar un vector de dimensión n para almacenar el número de veces que cada variable tomo el valor 0 o 1
* Si son enteras, se utiliza una matriz bidimensional como contador de las veces que la variable i toma el valor k
* Si son permutaciones de orden, se puede utilizar una matriz bidimensional como contador de las veces que el valor i ha ido seguido del j

Uso de la memoria de frecuencias:

* Generar directamente la nueva solución a partir de la información almacenada en esta memoria, dando mayor probabilidad de aparición a los valores menos habituales
* Usar la información almacenada para modificar temporalmente el caso del problema, potenciando los valores heurísticos de los atributos menos usados en la búsqueda.