

ALGORÍTMICA

2012 – 2013

- **Parte I. Introducción a las Metaheurísticas**
 - Tema 1. Metaheurísticas: Introducción y Clasificación
- **Parte II. Métodos Basados en Trayectorias y Entornos**
 - Tema 2. Algoritmos de Búsqueda Local Básicos
 - Tema 3. Algoritmos de Enfriamiento Simulado
 - **Tema 4. Algoritmos de Búsqueda Tabú**
 - Tema 5. Métodos Basados en Trayectorias Múltiples I: Métodos Multiarranque Básicos y GRASP
 - Tema 6. Métodos Basados en Trayectorias Múltiples II: ILS y VNS
- **Parte III. Métodos Basados en Poblaciones**
 - Tema 7. Algoritmos Genéticos
- **Parte IV. Intensificación y Diversificación**
 - Tema 8. Estudio del Equilibrio entre Intensificación y Diversificación
- **Parte V. Metaheurísticas Híbridas: Poblaciones y Trayectorias**
 - Tema 9. Algoritmos Meméticos
 - Tema 10. Modelos Híbridos II: *Scatter Search*
- **Parte VI. Paralelización de Metaheurísticas**
 - Tema 11. Metaheurísticas en Sistemas Descentralizados
- **Parte VII. Conclusiones**
 - Tema 12. Algunas Consideraciones sobre la Adaptación de Metaheurísticas a la Resolución de Problemas

ALGORÍTMICA

TEMA 4. Algoritmos de Búsqueda Tabú

1. INTRODUCCIÓN

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

3. ALGUNOS DATOS DE LA WEB SOBRE SU USO

4. EJEMPLO: VIAJANTE DE COMERCIO

5. APLICACIONES

- *A. Díaz y otros. Optimización Heurística y Redes Neuronales. Paraninfo, 1996*
- *F. Glover, B. Melián. Búsqueda Tabú. Inteligencia Artificial VII:2 (2003) 29-47.*
- *F. Glover, M. Laguna. Tabu Search. Kluwer Academic, 1997.*

1. INTRODUCCIÓN

Problemas de la Búsqueda Local (Tema 2)

Un mal diseño de la función objetivo puede guiar mal la búsqueda y dar lugar a que se obtengan soluciones de baja calidad

SOLUCIONES: 3 opciones para salir de los óptimos locales

- Permitir movimientos de **empeoramiento** de la solución actual (Ejemplo: Enfriamiento Simulado, Búsqueda Tabú, ...)
- Modificar la **estructura de entornos** (Ejemplo: Búsqueda Tabú, Búsqueda Descendente Basada en Entornos Variables: VND, Búsqueda en Entornos Variables: VNS, ...)
- Volver a **comenzar la búsqueda** desde otra solución inicial (Ejemplo: Búsquedas Multiarranque, ILS, Búsqueda Tabú, ...)

1. INTRODUCCIÓN

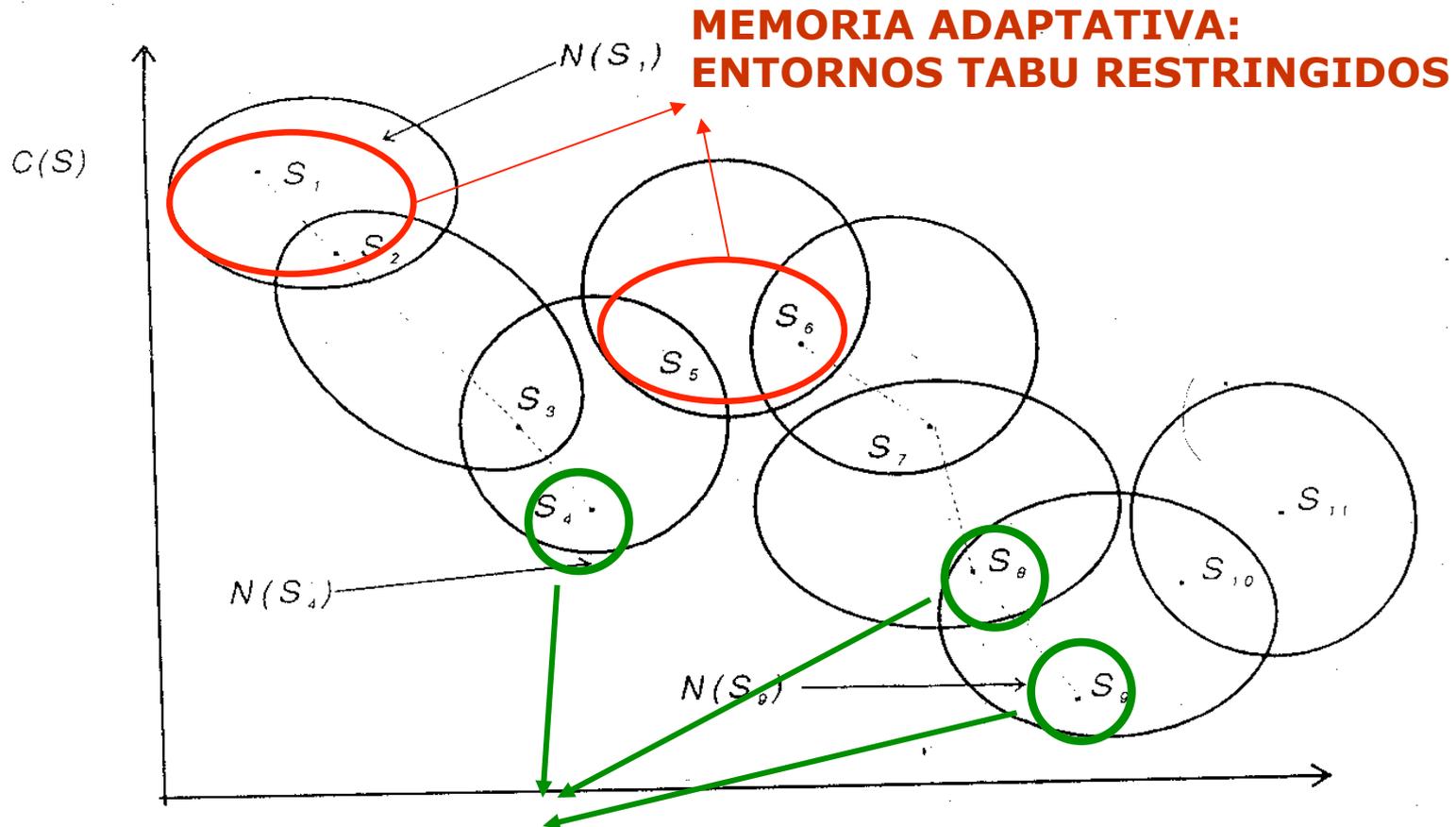
La Búsqueda Tabú es un procedimiento de búsqueda por entornos cuya característica distintiva es el uso de memoria adaptativa y estrategias especiales de resolución de problemas

La memoria adaptativa permite:

- restringir el entorno de búsqueda, e
- introducir mecanismos de reinicialización de la búsqueda mediante **intensificación** sobre zonas del espacio de búsqueda ya visitadas, o **diversificación** sobre posibles zonas del espacio de búsqueda poco visitadas

Glover, F. "Tabu Search — Part I", *ORSA Journal on Computing* 1989 1: 3, 190-206.
Glover, F. "Tabu Search — Part II", *ORSA Journal on Computing* 1990 2: 1, 4-32.

1. INTRODUCCIÓN



**MEMORIA ADAPTATIVA.
ESTRATEGIA ESPECIAL DE INTENSIFICACIÓN
SOBRE SOLUCIONES YA VISITADAS - Las mejores, distantes, ...**

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.1. Fundamentos

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

2.3. Ejemplo de Uso de la Memoria de Corto Plazo

2.4. Memoria de Largo Plazo: Intensificación y Diversificación de la Búsqueda

2.5. Ejemplo de Uso de la Memoria de Largo Plazo

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.1. 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 recorridos cíclicos, incorpora un mecanismo de generación de vecinos modificado que evita la exploración de zonas del espacio de búsqueda que ya han sido visitadas:

GENERACIÓN DE ENTORNOS TABÚ RESTRINGIDOS

- Emplea mecanismos de reinicialización para mejorar la capacidad del algoritmo para la exploración-explotación del espacio de búsqueda: **INTENSIFICACIÓN Y DIVERSIFICACIÓN**

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.1. Fundamentos

- Para realizar las dos tareas anteriores, hace uso de dos estructuras de memoria adaptativas distintas:
 - Memoria de corto plazo o Lista tabú
 - Memoria de largo plazo
- La memoria de corto plazo guarda información que permite guiar la búsqueda de forma inmediata, desde el comienzo del procedimiento (GENERACIÓN DE ENTORNOS TABÚ RESTRINGIDOS)

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.1. Fundamentos

- La 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 aplicando la memoria a corto plazo

La información guardada en esta memoria se usa para comenzar con la búsqueda desde otra solución inicial de acuerdo a dos filosofías distintas:

- Intensificar la búsqueda, volviendo a visitar zonas del espacio prometedoras (que contenían buenas soluciones), ya exploradas parcialmente
- Diversificar la búsqueda, visitando nuevas zonas no exploradas aún

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

Modificación de las Estructuras de Entorno

- La TS extiende la búsqueda local sustituyendo $E(S_{act})$ por otro entorno $E^*(S_{act}) \subset E(S_{act})$. En cada iteración, se acepta siempre el mejor vecino de dicho entorno, tanto si es peor como si es mejor que S_{act}
- La memoria de corto plazo (lista tabú) permite a la TS determinar $E^*(S_{act})$ y así organizar la manera en la que se explora el espacio

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

- Las soluciones admitidas en $E^*(S_{act})$ dependen de la estructura de la lista tabú:
 - **Lista de soluciones tabú:** Se identifican soluciones ya visitadas y se marcan como tabú para no volver a ellas, eliminándolas del vecindario de S_{act}
 - **Lista de movimientos tabú:** Se eliminan del entorno todos los vecinos resultantes de aplicar sobre S_{act} 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 contienen elementos tabú-activos se convierten en “soluciones tabú”

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

EJEMPLO:

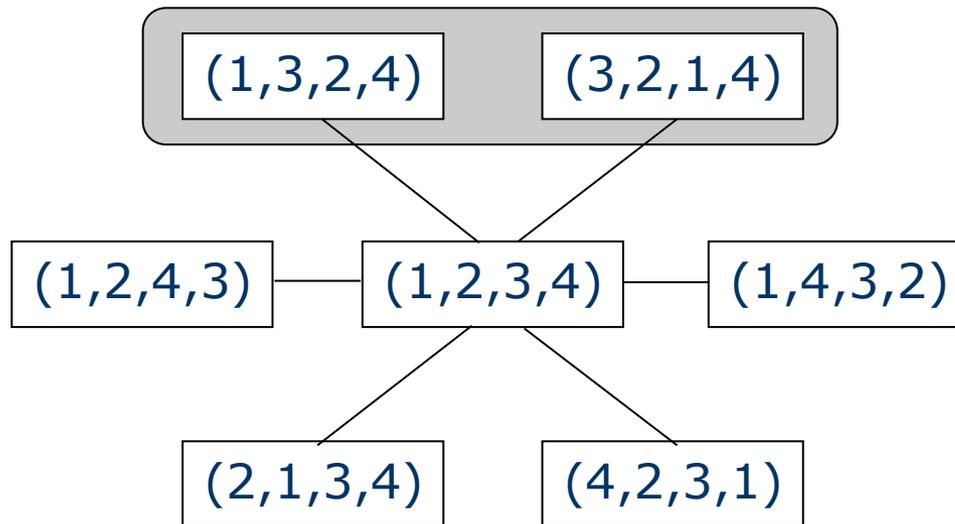
Lista tabú de soluciones:

$$LT = \{ (1,3,2,4), (3,1,2,4), (3,2,1,4) \}$$

Operador de vecino: 2-opt $S_{act} = (1,2,3,4)$

Vecindario reducido de S_{act} :

$$E^*(S_{act}) = \{ (2,1,3,4), \underline{(3,2,1,4)}, (4,2,3,1), \underline{(1,3,2,4)}, (1,4,3,2), (1,2,4,3) \}$$



2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

EJEMPLO:

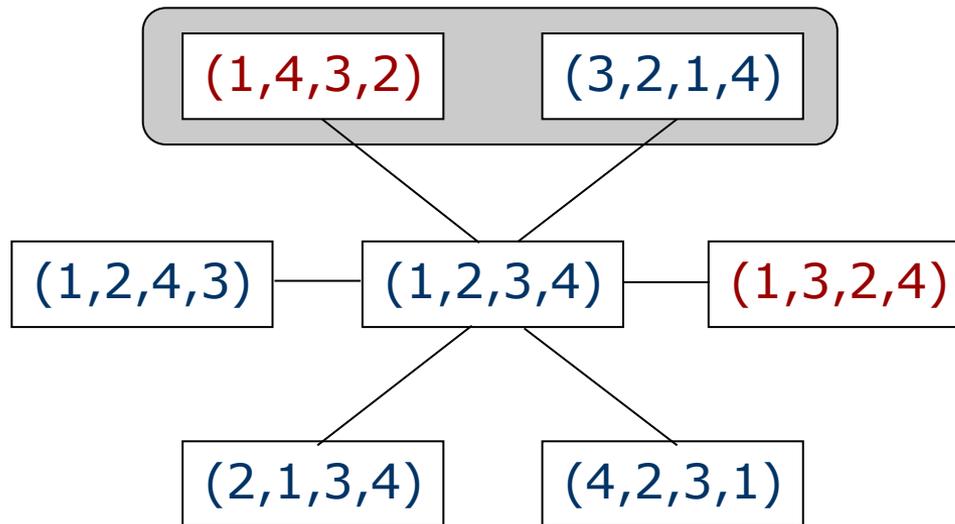
Lista tabú de movimientos:

$$LT = \{ (1,3), (2,4) \}$$

Operador de vecino: 2-opt $S_{act} = (1,2,3,4)$

Vecindario reducido de S_{act} :

$$E^*(S_{act}) = \{(2,1,3,4), \text{~~(3,2,1,4)~~, (4,2,3,1), (1,3,2,4), \text{~~(1,4,3,2)~~, (1,2,4,3) \}$$



2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

TENENCIA TABÚ

- Un atributo/movimiento o solución que se haya incluido en la lista tabú en algún momento de la búsqueda no permanece en ella para siempre
- Se denomina “tenencia tabú” al 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. Una vez transcurrido el valor especificado, el elemento en cuestión deja de ser tabú activo y se elimina de la lista

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

NIVELES DE ASPIRACIÓN

- El “criterio de aspiración” introduce un elemento importante de flexibilidad en la búsqueda tabú
- El estado tabú de un movimiento o atributo puede ser ignorado si se cumplen ciertas condiciones, en la forma de niveles de aspiración
- Por ejemplo, un vecino que sea mejor que cualquiera de las soluciones encontradas anteriormente merece ser considerado admisible, incluso aunque dicho vecino sea una solución tabú
- Una solución tabú dejará de serlo y se incluirá en el entorno factible si supera un cierto nivel de aspiración

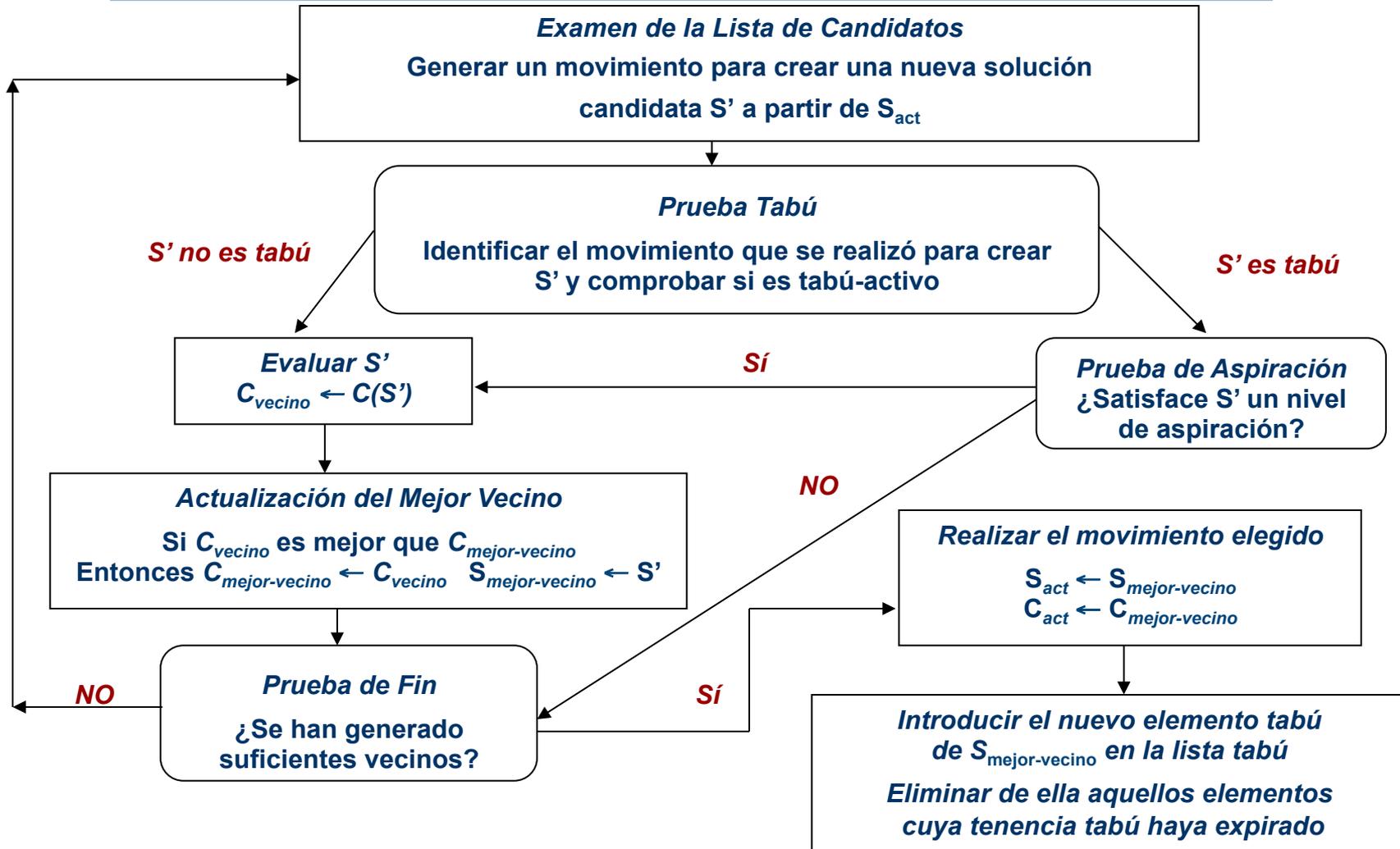
2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

ESTRATEGIAS PARA LA LISTA DE CANDIDATOS

- Las estrategias para la lista de candidatos se usan para restringir el número de vecinos examinados en una iteración dada, para los casos en los que $E^*(S_{act})$ es grande o la evaluación de sus elementos es costosa
- Se busca el mejor movimiento disponible que pueda ser determinado con una cantidad apropiada de esfuerzo
- Así, no se genera el entorno reducido completo sino una parte del mismo y se toma el mejor vecino. La selección adecuada de candidatos puede reducir apreciablemente los tiempos de ejecución

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ



2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

ESTRUCTURAS DE MEMORIA DE CORTO PLAZO

- Sea $I = \{1, 2, \dots, n\}$ un conjunto de índices para una colección de atributos y sea S una solución al problema
- Sea $i \in I$, el estado tabú-activo de un atributo S_i se representa por el índice i , que puede almacenarse en un vector o lista
- El estado tabú de $S_i = k$ se representa guardando el par ordenado $\langle i, k \rangle$ en la lista
- Para el caso de un intercambio de dos índices $i, j \in I$, se guardaría el par $\langle i, j \rangle$

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

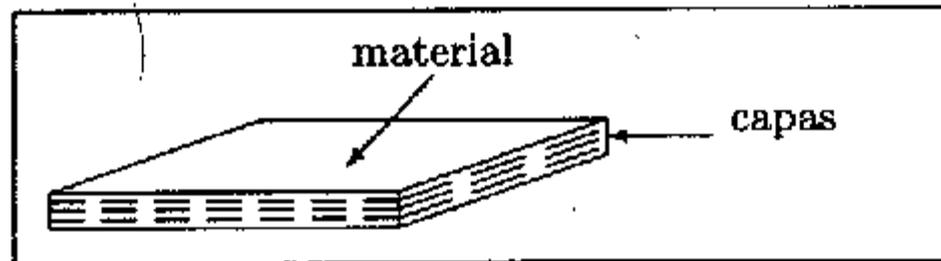
2.2. Memoria de Corto Plazo: Generación de Entornos Restringidos en la Búsqueda Tabú

- Para facilitar la gestión de la tenencia tabú de los atributos (el número de iteraciones que son tabú-activos), 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
- Así, [la tenencia tabú de todos los atributos equivale al tamaño de la lista](#). La elección de un valor para la tenencia tabú suele ser experimental, siendo una función del número total de atributos (p.e., $[n/3, 3 \cdot n]$)

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

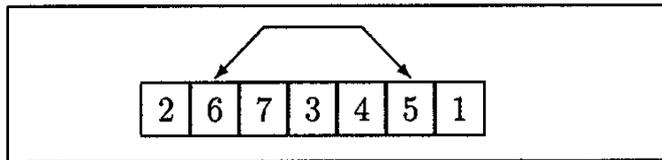
2.3. Ejemplo de Uso de la Memoria de Corto Plazo: Diseño de un Material Formado por un Número de Capas Aislantes

- **Problema de permutaciones** que consiste en encontrar el orden de las capas que maximiza el valor de aislamiento total del material compuesto. Supongamos que se consideran 7 capas para un material particular y que evaluar el valor de aislamiento total de una ordenación particular es un procedimiento computacionalmente costoso. Maximizamos una función de aislamiento

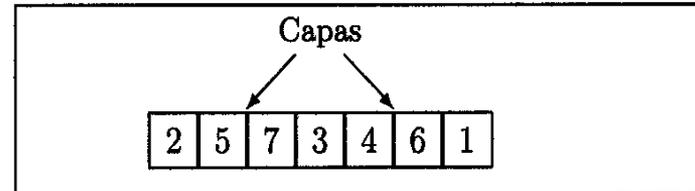


2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

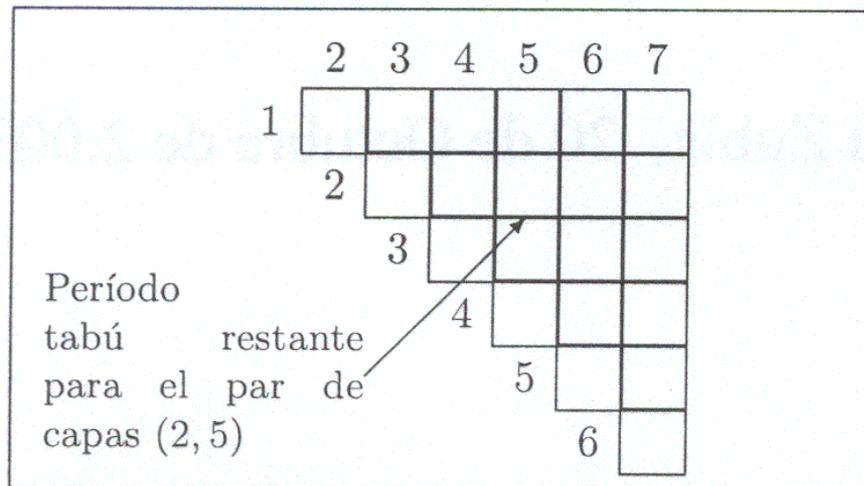
2.3. Ejemplo de Uso de la Memoria de Corto Plazo



Movimiento - Permutación



Intercambio de las capas 5 y 6

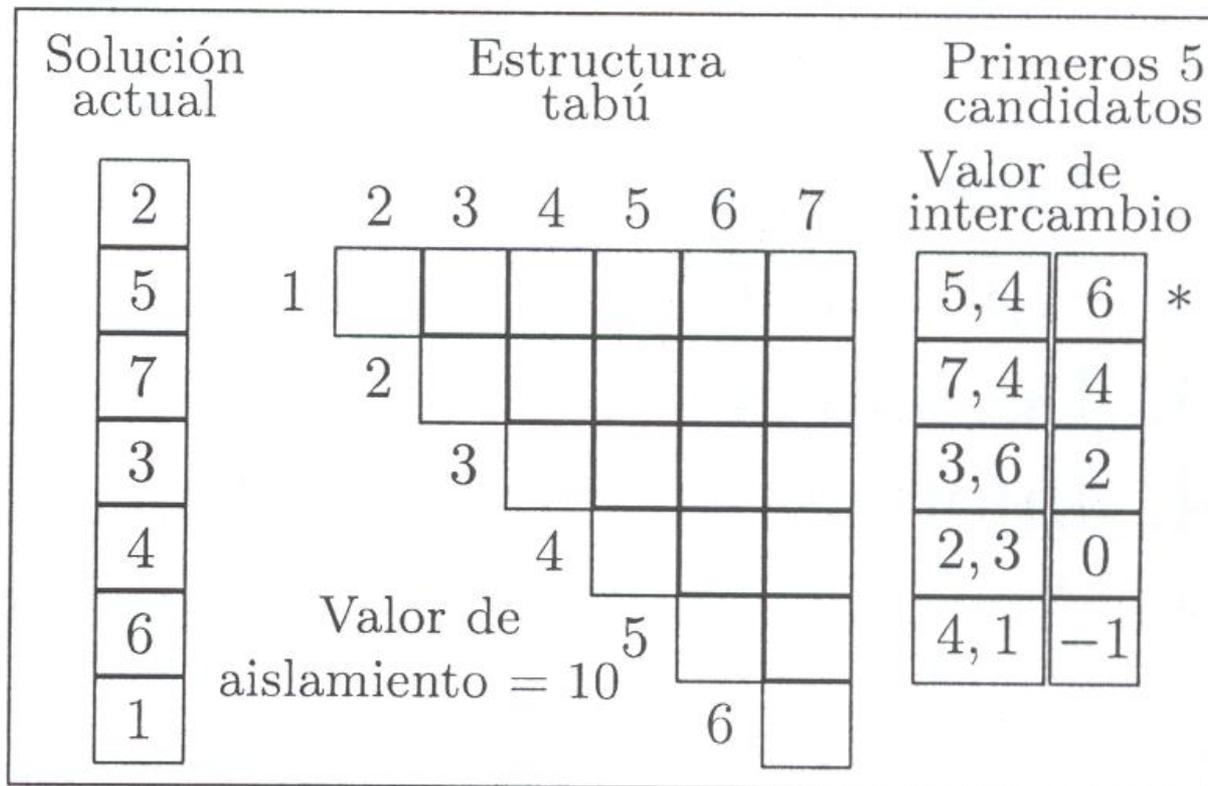


Estructura de datos de la Lista Tabú

Se impide el intercambio de las capas i y j mientras el movimiento sea tabú-activo

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

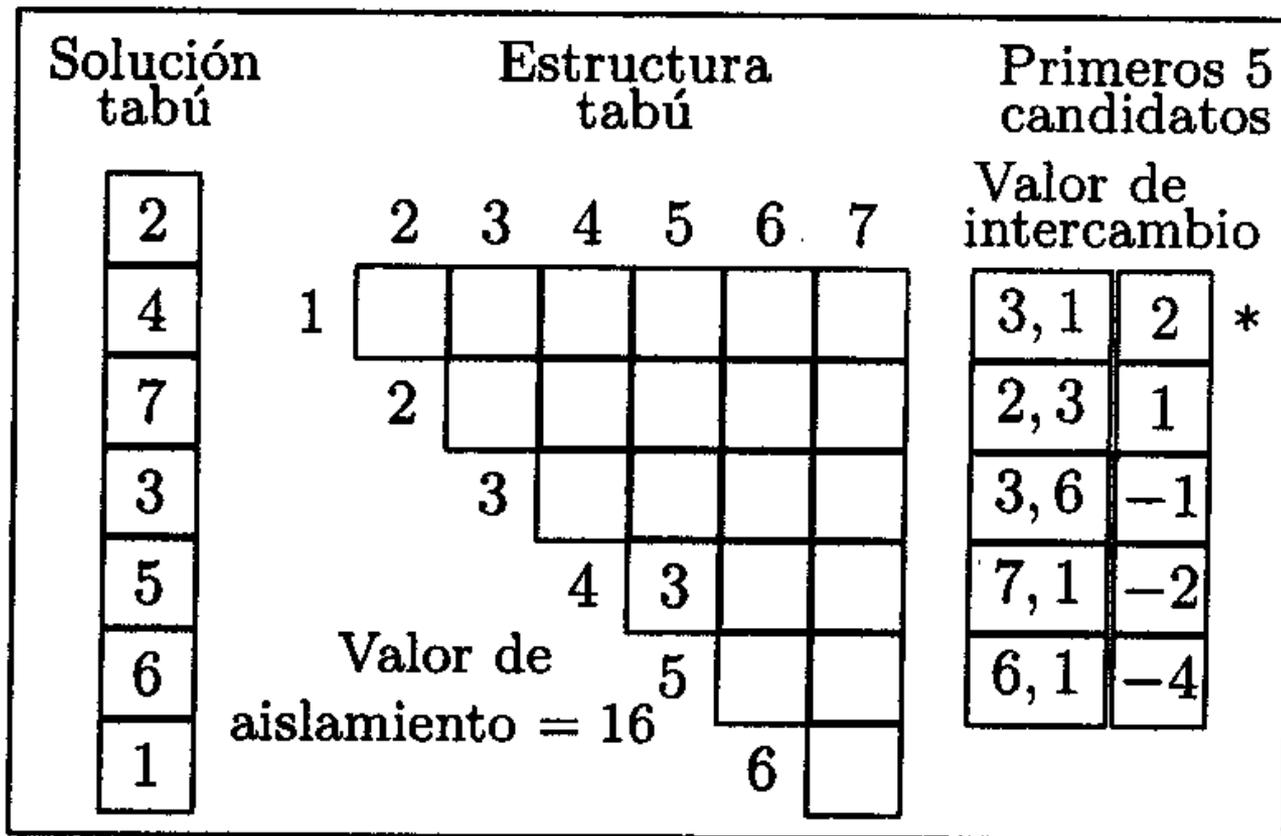
2.3. Ejemplo de Uso de la Memoria de Corto Plazo



Iteración 0

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

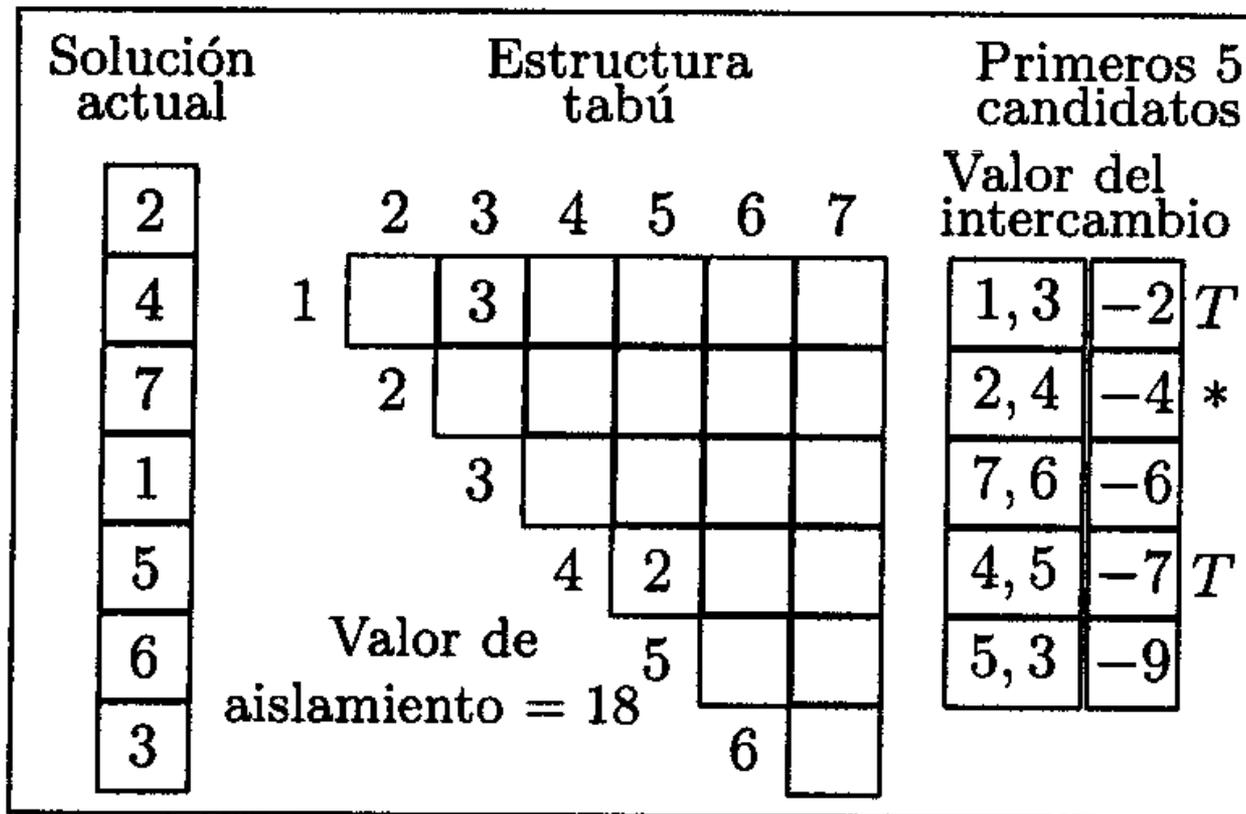
2.3. Ejemplo de Uso de la Memoria de Corto Plazo



Iteración 1 (Tenencia Tabú = 3)

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

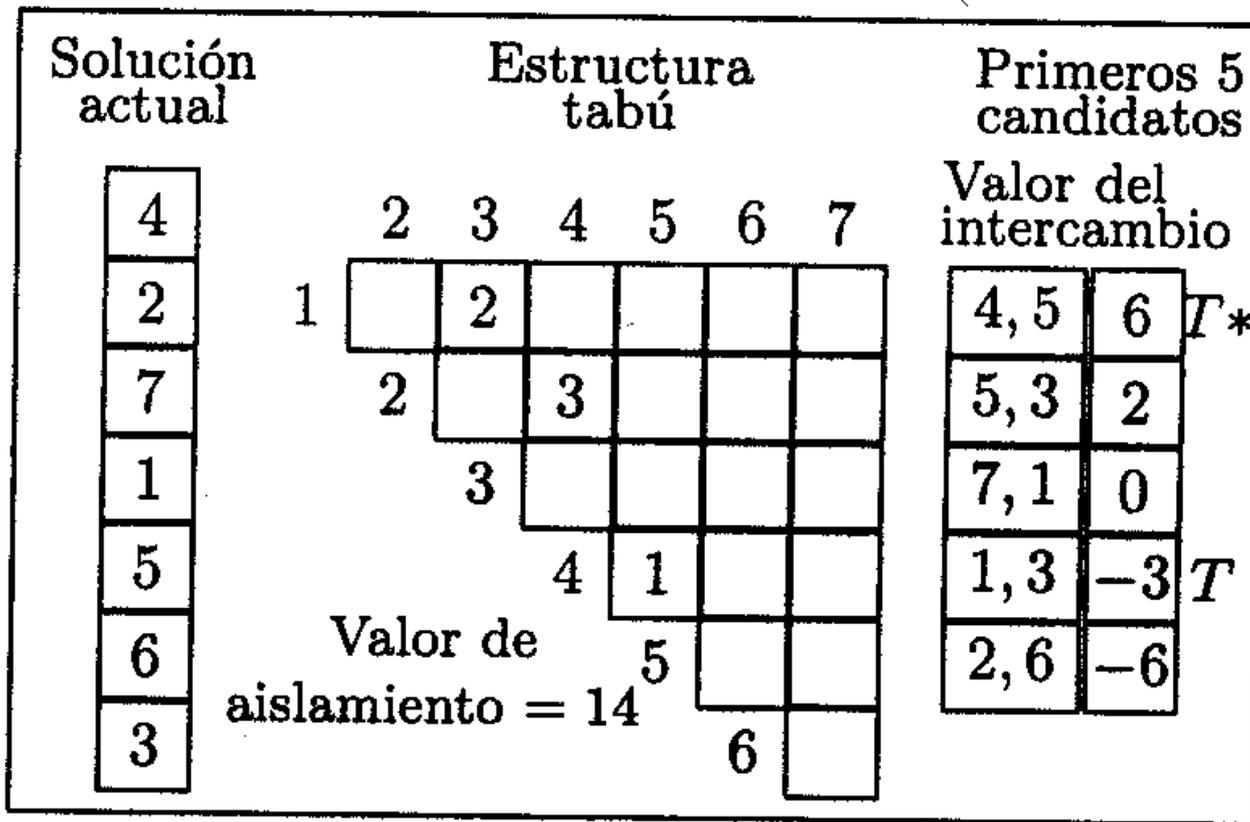
2.3. Ejemplo de Uso de la Memoria de Corto Plazo



Iteración 2

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

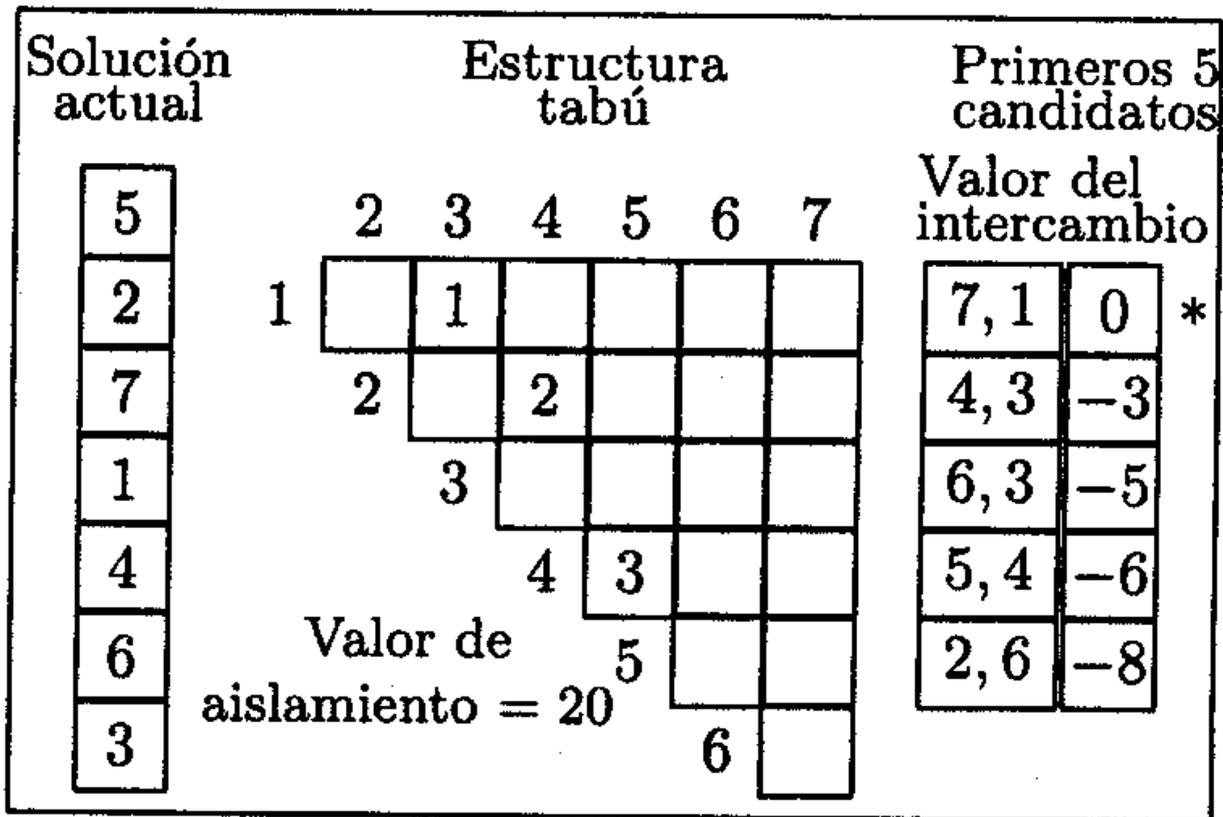
2.3. Ejemplo de Uso de la Memoria de Corto Plazo



Iteración 3

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.3. Ejemplo de Uso de la Memoria de Corto Plazo



Iteración 4

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

2.4. Memoria de Largo Plazo: Intensificación y Diversificación de la Búsqueda

- En algunas aplicaciones, las componentes de la memoria TS de corto plazo son suficientes para producir soluciones de muy alta calidad
- No obstante, en general, la TS se vuelve mucho más potente incluyendo memoria de largo plazo y sus estrategias de reinicialización asociadas
- La memoria de largo plazo se puede usar de dos modos distintos:
 - Estrategias de Intensificación
 - Estrategias de Diversificación
- 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

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

ESTRATEGIAS DE INTENSIFICACIÓN

- Se basan en una reinicialización de la búsqueda que efectúa un regreso a regiones atractivas del espacio para buscar en ellas más extensamente
- Se mantiene un registro de las mejores soluciones visitadas, insertando una nueva solución cada vez que se convierte en la mejor global
- Se puede introducir una medida de diversificación para asegurar que las soluciones registradas difieran una de otra en un grado deseado

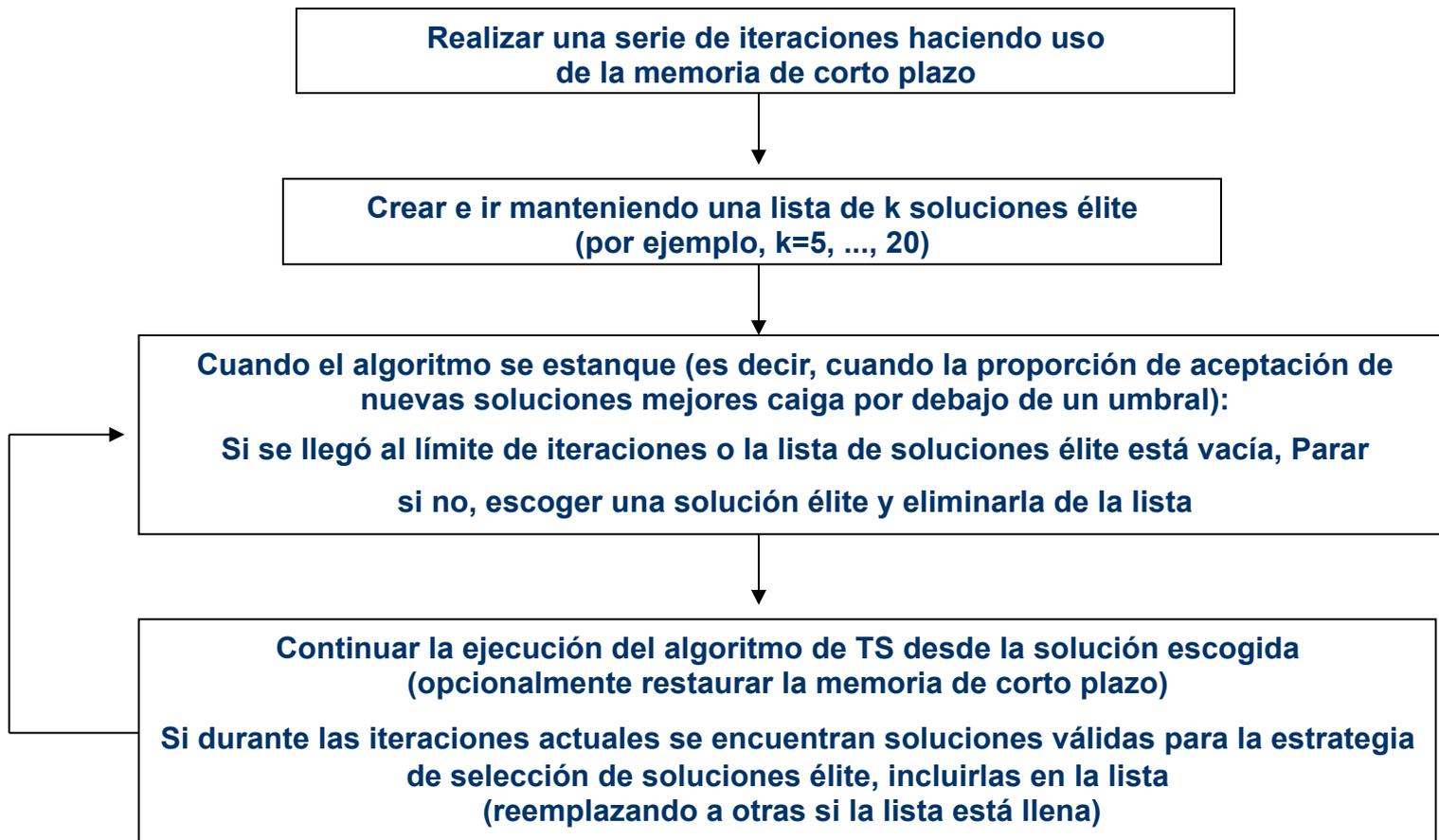
Tenemos cuatro variantes resultantes de combinar “soluciones/memoria”:

- **Solución desde la que se reinicializa:**
 - Reanudar el proceso *desde la mejor* de las soluciones registradas
 - Usar una pila de mejores soluciones de longitud limitada y escoger *la cabeza de la pila*
- **Restauración de la memoria de corto plazo:**
 - *Almacenar la memoria de corto plazo en el momento que se encontró cada una de las mejores soluciones para restaurarla cuando se reanude la búsqueda desde dicha solución*
 - *Borrar la memoria de corto plazo para iniciar la búsqueda desde cero*

2.4. Memoria de Largo Plazo: Intensificación y Diversificación de la Búsqueda

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

Enfoque simple de intensificación en TS



2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

ESTRATEGIAS DE DIVERSIFICACIÓN

- Conducen la búsqueda hacia nuevas regiones del espacio de búsqueda no exploradas aún
- La búsqueda se reinicializa cuando se estanca, partiendo de una solución no visitada
- Esta solución se genera a partir de la memoria de frecuencias, dando mayor probabilidad de aparición a los valores menos habituales

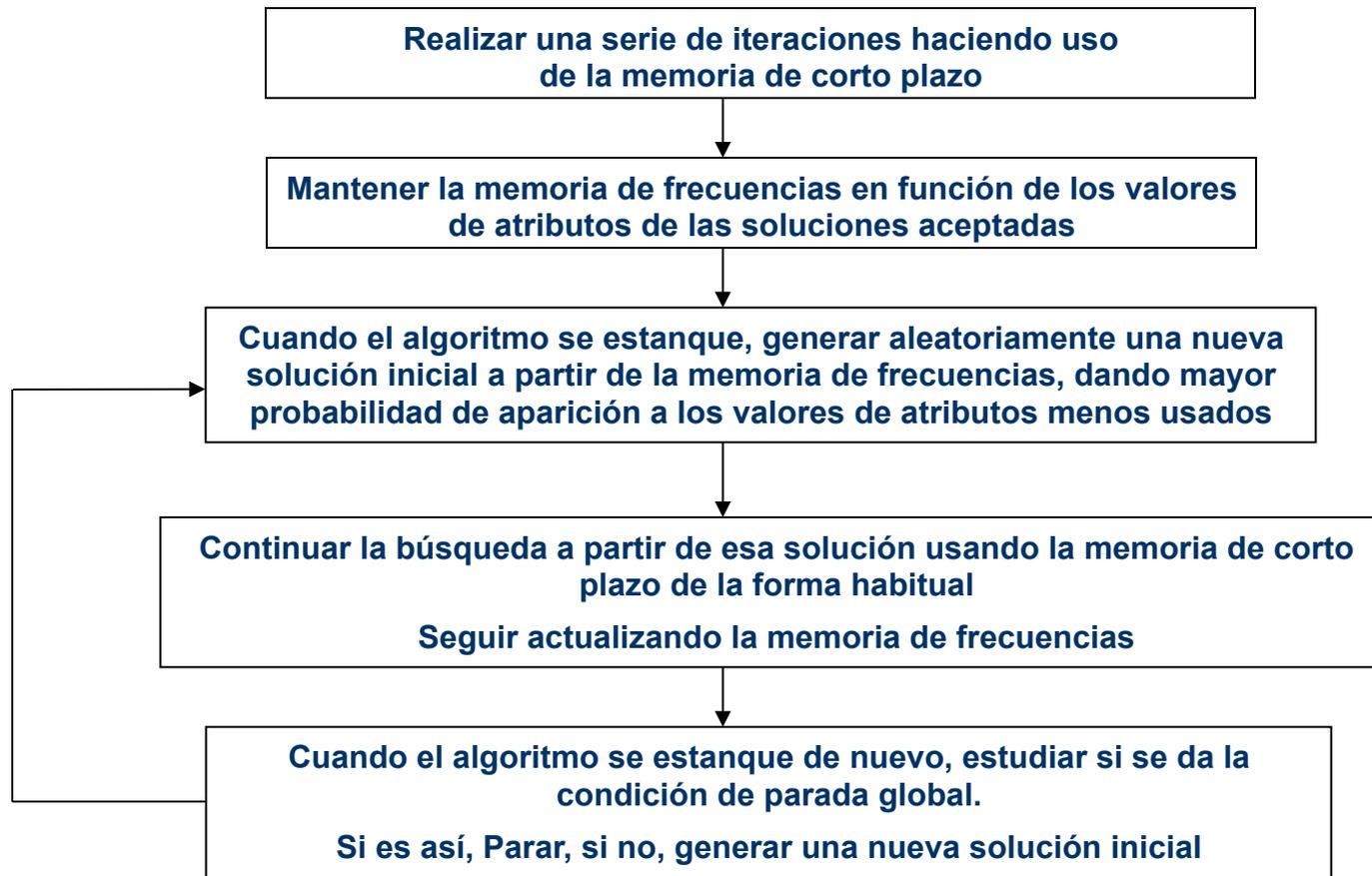
OSCILACIÓN ESTRATÉGICA

- Interacción efectiva entre intensificación y diversificación consistente en escoger aleatoriamente una u otra cuando la búsqueda se estanque

2.4. Memoria de Largo Plazo: Intensificación y Diversificación de la Búsqueda

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

Enfoque simple de diversificación en TS



2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

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 tomó el valor 0 (ó 1)
- Si son enteras, se utiliza una matriz bidimensional, como contador de las veces que la variable i toma el valor k : $M[i,k]$
- 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 : $M[i,j]$

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

USO DE LA MEMORIA DE FRECUENCIAS

■ **Existen dos posibilidades:**

- 1) Generar directamente la nueva solución inicial a partir de la información almacenada en la memoria de frecuencias **M**, dando mayor probabilidad de aparición a los valores menos habituales**
- 2) Usar la información almacenada en **M** para modificar temporalmente el caso del problema, potenciando los valores heurísticos de los atributos menos usados en la búsqueda**

Aplicar un algoritmo greedy sobre ese caso modificado para generar la solución inicial. Restaurar el caso original del problema antes de continuar con la búsqueda

2. LA ESTRUCTURA DE LA BÚSQUEDA TABÚ

Solución actual		Estructura tabú (Reciente)						
		1	2	3	4	5	6	7
1								
3	1				3			
6	2							
2	3	3					2	
7	4	1	5					1
5	5		4		4			
4	6			1		2		
	7	2			3			

Valor de aislamiento = 12 (Frecuente)

Problema: Diseño de un Material Formado por un Número de Capas Aislantes

Ejemplo de estructura de memoria con diagonal superior como lista tabú, y diagonal inferior midiendo la frecuencia de aparición de las capas del material en las posiciones correspondientes, para poder diversificar la búsqueda en futuras reinicializaciones

Al reinicializar, se generaría una nueva solución en la que en cada posición se le daría más probabilidad a la capa que menos frecuentemente la hubiera ocupado en el pasado

**Estructura de Memoria Complementaria:
Memoria a Largo Plazo**

3. EJEMPLO: VIAJANTE DE COMERCIO

- 1. Generación de la solución inicial:** aleatoria
- 2. Esquema de representación:** Representación de orden mediante permutación $\{1, \dots, n\}$
- 3. Operador de generación de vecinos:** seleccionar dos ciudades e intercambiarlas (2-opt)
- 4. Función objetivo (minimización):**

$$C(S) = \sum_{i=1}^{n-1} (D[S[i], S[i+1]]) + D[S[n], S[1]]$$

Los vecinos se evalúan con el cálculo optimizado de la función objetivo:

$$C(S') = C(S) - D(S[i-1], S[i]) - D(S[i], S[i+1]) - D(S[j-1], S[j]) - D(S[j], S[j+1]) + D(S[i-1], S[j]) + D(S[j], S[i+1]) + D(S[j-1], S[i]) + D(S[i], S[j+1])$$

3. EJEMPLO: VIAJANTE DE COMERCIO

5. **Lista tabú (L)**: Cada vez que se acepta una solución, se almacena en la lista tabú el movimiento que provocó dicha solución

Para ello, se almacenan las dos ciudades intercambiadas y las nuevas posiciones que ocupan $(i, j, \text{Pos}(i), \text{Pos}(j))$. No se permiten intercambios que den lugar a que la ciudad i ocupe la posición $\text{Pos}(i)$, o la ciudad j la posición $\text{Pos}(j)$

Ejemplo:

1	2	3	4	5	6	7	8
(1	2	<u>4</u>	3	8	<u>5</u>	7	6)
(1	2	<u>5</u>	3	8	<u>4</u>	7	6)

$$L = L - \{\text{movimiento más antiguo}\} + \{(4,5,6,3)\}$$

Para decidir si un movimiento es tabú activo, hay que tener en cuenta que $(i, j, \text{Pos}(i), \text{Pos}(j)) = (j, i, \text{Pos}(j), \text{Pos}(i))$

3. EJEMPLO: VIAJANTE DE COMERCIO

6. **Estrategia de selección de vecinos:** Examinar 50 movimientos (todos distintos) y escoger el mejor de acuerdo a los criterios tabú
7. **Criterio de aspiración:** Tener menor coste que la mejor solución obtenida hasta el momento
8. **Estrategias de reinicialización:**
 - a) Generar una solución aleatoria y continuar el algoritmo a partir de ella (**diversificación**)
 - b) Volver a la mejor solución obtenida hasta el momento y continuar el algoritmo a partir de ella (**intensificación**)
 - c) Generar una nueva solución *greedy* a partir de la memoria a largo plazo y continuar el algoritmo a partir de ella (**diversificación controlada**)

3. EJEMPLO: VIAJANTE DE COMERCIO

- **Memoria a largo plazo:** Se usa una matriz simétrica *frec* que almacena el número de veces que cada par de ciudades han estado consecutivas en alguna de las soluciones aceptadas durante la búsqueda

Para reinicializar provocando diversidad, se aplica el algoritmo *greedy* sobre una matriz de distancias modificada de forma que valores altos de frecuencia entre un par de ciudades supongan un valor alto de distancia entre ellas:

$$d(i, j) = d(i, j) + \mu \cdot (d_{\max} - d_{\min}) \cdot \frac{frec(i, j)}{frec_{\max}}$$

siendo d_{\max} y d_{\min} la mayor y menor distancia existente entre dos ciudades, y $frec_{\max}$ la frecuencia más alta existente en la matriz de memoria a largo plazo. El parámetro μ dicta el grado de alteración de la distancia (p.e., $\mu=0,3$)

La alteración de distancias sólo se usa para generar la solución diversa con el *greedy* y no afecta a la función objetivo en la búsqueda posterior

3. EJEMPLO: VIAJANTE DE COMERCIO

- Hay distintas posibilidades para la lista tabú del algoritmo de BT para el Viajante de Comercio. La elección de una de ellas da lugar a un algoritmo distinto:
 1. **Vector $(i,j,Pos(i),Pos(j))$** : Vector que evita cualquier intercambio que dé lugar a una solución en la que las ciudades i y j ocupen las posiciones $Pos(i)$ y $Pos(j)$ respectivamente
 2. **Vector $(i,j,Pos(i),Pos(j))$** : Mismo vector que evita un intercambio que dé lugar a una solución en la que la ciudad i ocupe la posición $Pos(i)$ y la j ocupe la posición $Pos(j)$
 3. **Vector $(i,Pos(i))$** : Para evitar que la ciudad i vuelva a la posición $Pos(i)$
 4. **Ciudad i** : Para evitar que la ciudad i se mueva a la izquierda de su posición actual

3. EJEMPLO: VIAJANTE DE COMERCIO

5. Ciudad i : Para evitar que la ciudad i se mueva (cambie su posición)
 6. Vector $(j, \text{Pos}(j))$: Para evitar que la ciudad j vuelva a la posición $\text{Pos}(j)$
 7. Ciudad j : Para evitar que la ciudad j se mueva a la derecha de su posición actual
 8. Ciudad j : Para evitar que la ciudad j se mueva (cambie su posición)
 9. Ciudades i y j : Para evitar que ambas ciudades cambien su posición actual
- En las condiciones 3 a 9 se supone que las ciudades i y j están situadas de tal forma que $\text{Pos}(i) < \text{Pos}(j)$. La condición 1 es la menos restrictiva y la condición 9 la más restrictiva. El nivel restrictivo de las condiciones 3, 4 y 5 es creciente

4. APLICACIONES: MDP

- **Generación de la solución inicial:** aleatoria

- **Esquema de representación:**

$$S = \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \boxed{\textcircled{0}} & \boxed{1} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{\textcircled{1}} \end{array}$$

- **Operador de generación de vecinos:** Escoger un elemento seleccionado y otro no seleccionado e intercambiar sus estados.
- **Función objetivo (maximización):**

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_i x_j$$

4. APLICACIONES: SI (II)

- 5. Lista tabú:** Cada vez que se acepta una solución, se almacena en la lista tabú el movimiento que provocó dicha solución. Tenencia tabú inicial: 50

Para ello, se almacena la instancia cuyo estado va a ser cambiado

Ejemplo:

1	2	3	4	5	6	7	8
(0	1	<u>1</u>	0	0	1	1	0)
(0	1	<u>0</u>	0	0	1	1	0)

$$L = L - \{\text{instancia más antigua}\} + \{3\}$$

4. APLICACIONES: SI (III)

6. **Estrategia de selección de vecino**: Examinar 30 vecinos y escoger el mejor de acuerdo a los criterios tabú
7. **Criterio de aspiración**: Tener mayor coste que la mejor solución obtenida hasta el momento
8. **Estrategias de reinicialización**:
 - Generar una solución aleatoria y continuar el algoritmo a partir de ella (**diversificación**)
 - Volver a la mejor solución obtenida hasta el momento y continuar el algoritmo a partir de ella (**intensificación**)
 - Generar una nueva solución poco frecuente a partir de la memoria a largo plazo y continuar el algoritmo a partir de ella (**diversificación controlada**)

4. APLICACIONES: SI (IV)

- **Memoria a largo plazo:** Se usa un vector *frec* que almacena el número de veces que cada objeto ha pertenecido al subconjunto *S* en alguna de las soluciones aceptadas durante la búsqueda

Para reinicializar provocando diversidad, se incluirá o no la instancia *j* en *S* dando menor probabilidad de ser incluida cuanto mayor valor tenga en *frec_j*. Esta probabilidad es:

$$\text{Instancia } j \left\{ \begin{array}{l} \in S, \quad \text{si } u < 1 - \frac{frec_j}{num_soluciones} \\ \notin S, \quad \text{si } u \geq 1 - \frac{frec_j}{num_soluciones} \end{array} \right.$$

donde *u* es un valor aleatorio uniforme en $[0,1]$

4. APLICACIONES: SI (V)

- Se realizan 4 reinicializaciones (separadas por un número de iteraciones constante) durante el desarrollo del algoritmo
- Cada vez que se va a realizar una reinicialización, se escoge una de las tres opciones según la siguiente distribución de probabilidades:
 - 0,25: Nueva solución actual aleatoria
 - 0,25: Volver a la mejor solución generada
 - 0,50: Solución poco frecuente a partir de la memoria a largo plazo
- Independientemente de la reinicialización realizada, se modifica el tamaño de la lista tabú para provocar un cambio más efectivo en el comportamiento del algoritmo. El tamaño de cada lista se incrementa o decrementa en un 50% (modificación sobre el tamaño actual independiente para cada lista)
- Tras la reinicialización, las listas de corto plazo se inicializan a cero, mientras que la de largo plazo no se modifica

ALGORÍTMICA

2012 - 2013

- **Parte I. Introducción a las Metaheurísticas**
 - Tema 1. Metaheurísticas: Introducción y Clasificación
- **Parte II. Métodos Basados en Trayectorias y Entornos**
 - Tema 2. Algoritmos de Búsqueda Local Básicos
 - Tema 3. Algoritmos de Enfriamiento Simulado
 - Tema 4. Algoritmos de Búsqueda Tabú
 - Tema 5. Métodos Basados en Trayectorias Múltiples I: Métodos Multiarranque Básicos y GRASP
 - Tema 6. Métodos Basados en Trayectorias Múltiples II: ILS y VNS
- **Parte III. Métodos Basados en Poblaciones**
 - Tema 7. Algoritmos Genéticos
- **Parte IV. Intensificación y Diversificación**
 - Tema 8. Estudio del Equilibrio entre Intensificación y Diversificación
- **Parte V. Metaheurísticas Híbridas: Poblaciones y Trayectorias**
 - Tema 9. Algoritmos Meméticos
 - Tema 10. Modelos Híbridos II: *Scatter Search*
- **Parte VI. Paralelización de Metaheurísticas**
 - Tema 11. Metaheurísticas en Sistemas Descentralizados
- **Parte VII. Conclusiones**
 - Tema 12. Algunas Consideraciones sobre la Adaptación de Metaheurísticas a la Resolución de Problemas