

Metaheurísticas

Unidad 2
Metaheurísticas basadas en Trayectorias

Tema 2: Búsqueda tabú

Objetivos

- Conocer los fundamentos de las búsquedas basadas en trayectorias como la búsqueda tabú
- Distinguir el funcionamiento de la memoria de corto plazo y de la memoria de largo plazo, así como el cometido de cada una de ellas
- Tener capacidad para, dado un problema dado, saber aplicar las técnicas de Búsqueda Tabú para su resolución

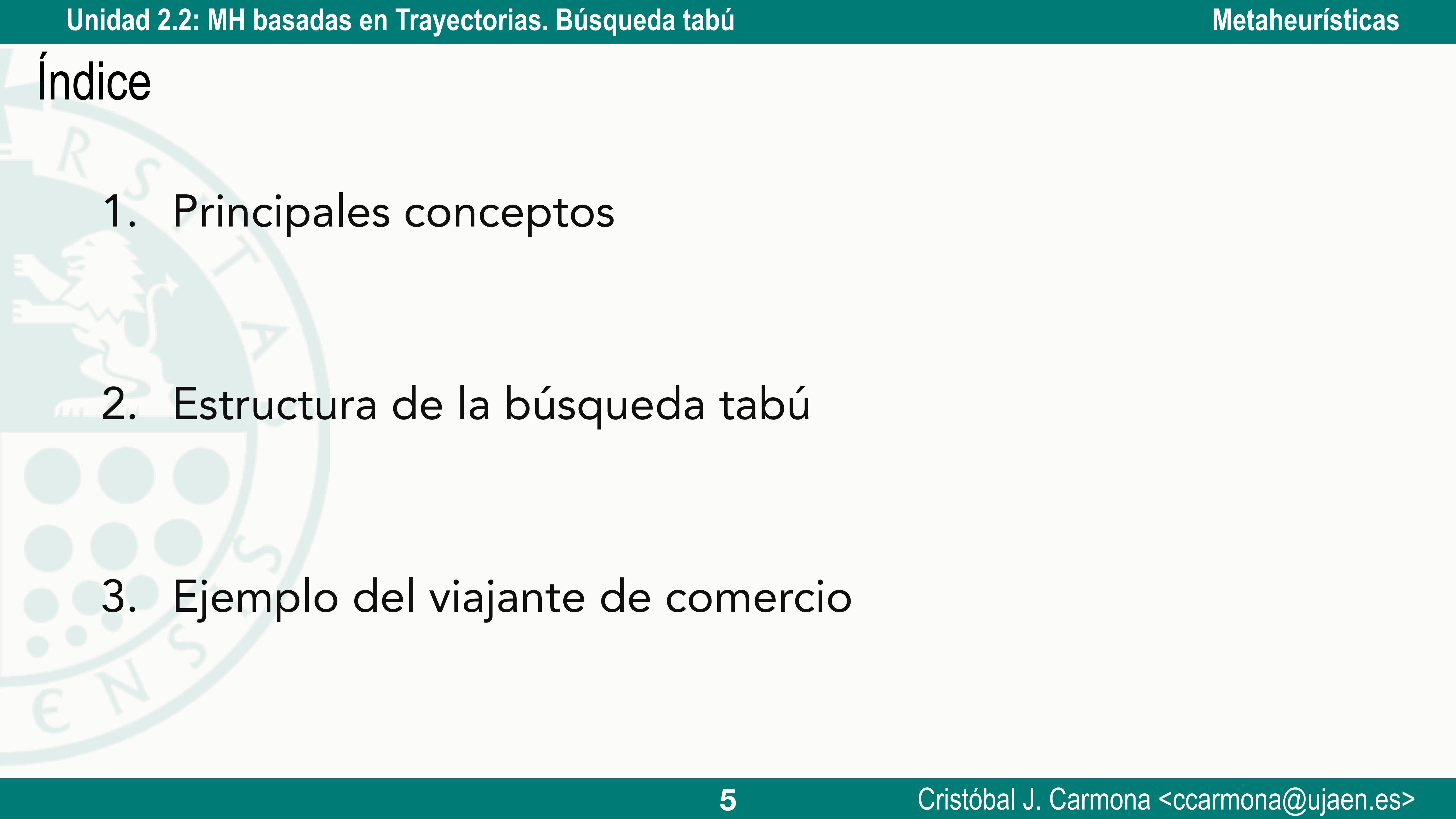
Bibliografía

- [Diaz96] A. Díaz y otros. Optimización Heurística y Redes Neuronales. Paraninfo, 1996
- [Glo03] F. Glover, B. Melián. Búsqueda Tabú. Inteligencia Artificial VII:2 (2003) 29-47.
- [Glo97] F. Glover, M. Laguna. Tabu Search. Kluwer Academic, 1997.
- [Gen03] M. Gendreau. Chapter 2: An Introduction to Tabu Search. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academic. (2003) 37-54.

Motivación

- Un mal diseño de la función objetivo puede guiar mal la búsqueda y dar lugar a que se obtengan soluciones malas
- Permitir movimientos de empeoramiento de la solución actual (Ejemplo: Enfriamiento Simulado, **Búsqueda Tabú**, ...)
- Modificar la estructura de entornos (Ejemplo: **Búsqueda Tabú**, Búsqueda en Entornos Variables: VNS, ...)
- Volver a comenzar la búsqueda desde otra solución inicial (Ejemplo: Búsquedas Multiarranque, ILS, **Búsqueda Tabú**, ...)

Índice

- 
1. Principales conceptos
 2. Estructura de la búsqueda tabú
 3. Ejemplo del viajante de comercio

Principales conceptos

introducción

El término Tabú proviene de Tongan, un lenguaje de la Polinesia utilizado por los aborígenes de la isla Tonga

- Hacía referencia a un objeto que no podía tocarse porque era sagrado

Según la RAE:

- Prohibición de comer o tocar algún objeto, impuesta a sus adeptos por algunas religiones de la Polinesia

Principales conceptos

introducción

- Se entiende que es una **prohibición impuesta por el entorno social como una medida de protección** o algo que se considera que tiene un riesgo
- Cualquier prohibición por protección **puede obviarse si lo demanda la ocasión**
- El tabú o la búsqueda tabú puede evitarse si hay evidencias de que las alternativas son competitivas

Principales conceptos

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

- 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

Principales conceptos

introducción

La memoria adaptativa permite:

- **restringir el entorno de búsqueda**, y
- **explorar con capacidad de reacción** (*responsive exploration*) con 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

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 recorridos cílicos, 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**

Estructura de la búsqueda tabú

fundamentos

Para realizar las dos tareas anteriores, hace uso de dos estructuras de memoria adaptativas distintas



**Memoria de corto plazo
(lista tabú)**

Memoria de largo plazo

Estructura de la búsqueda tabú

fundamentos

**Memoria de corto plazo
(lista tabú)**



Guarda información que permite guiar la **búsqueda de forma inmediata**, desde el comienzo del procedimiento

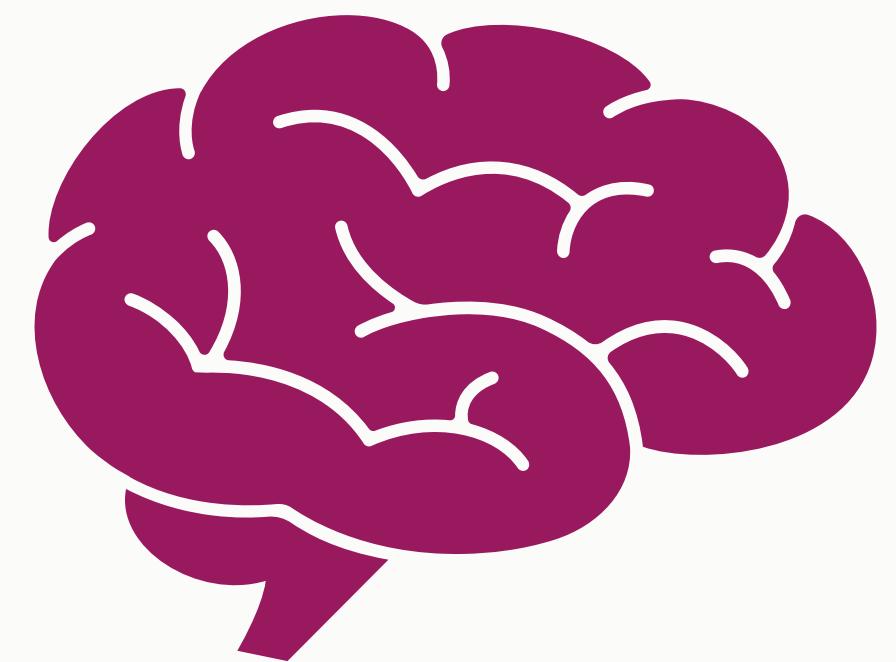
GENERACIÓN DE ENTORNOS TABÚ RESTRINGIDOS

Estructura de la búsqueda tabú

fundamentos

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



Estructura de la búsqueda tabú

fundamentos

Memoria de largo 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



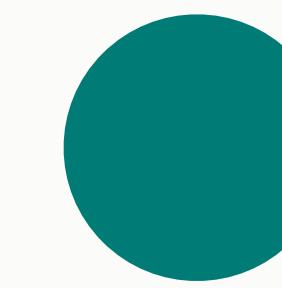
Estructura de la búsqueda tabú

fundamentos



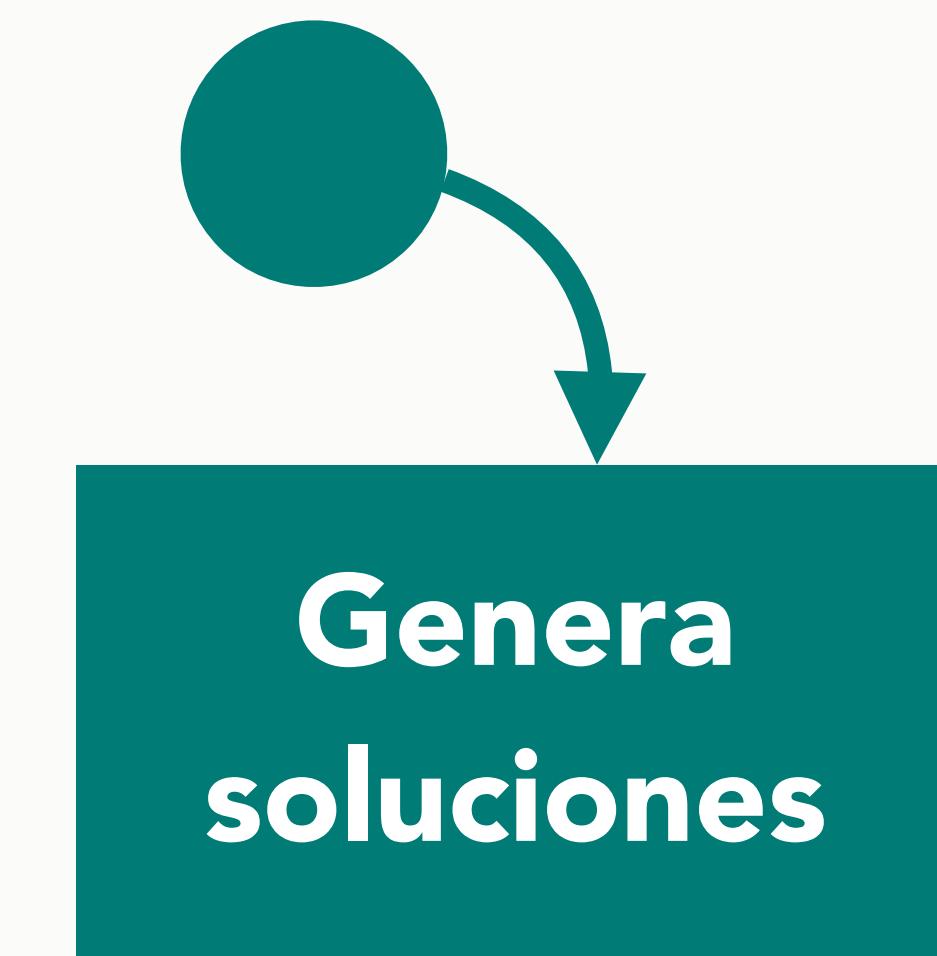
Estructura de la búsqueda tabú

fundamentos



Estructura de la búsqueda tabú

fundamentos



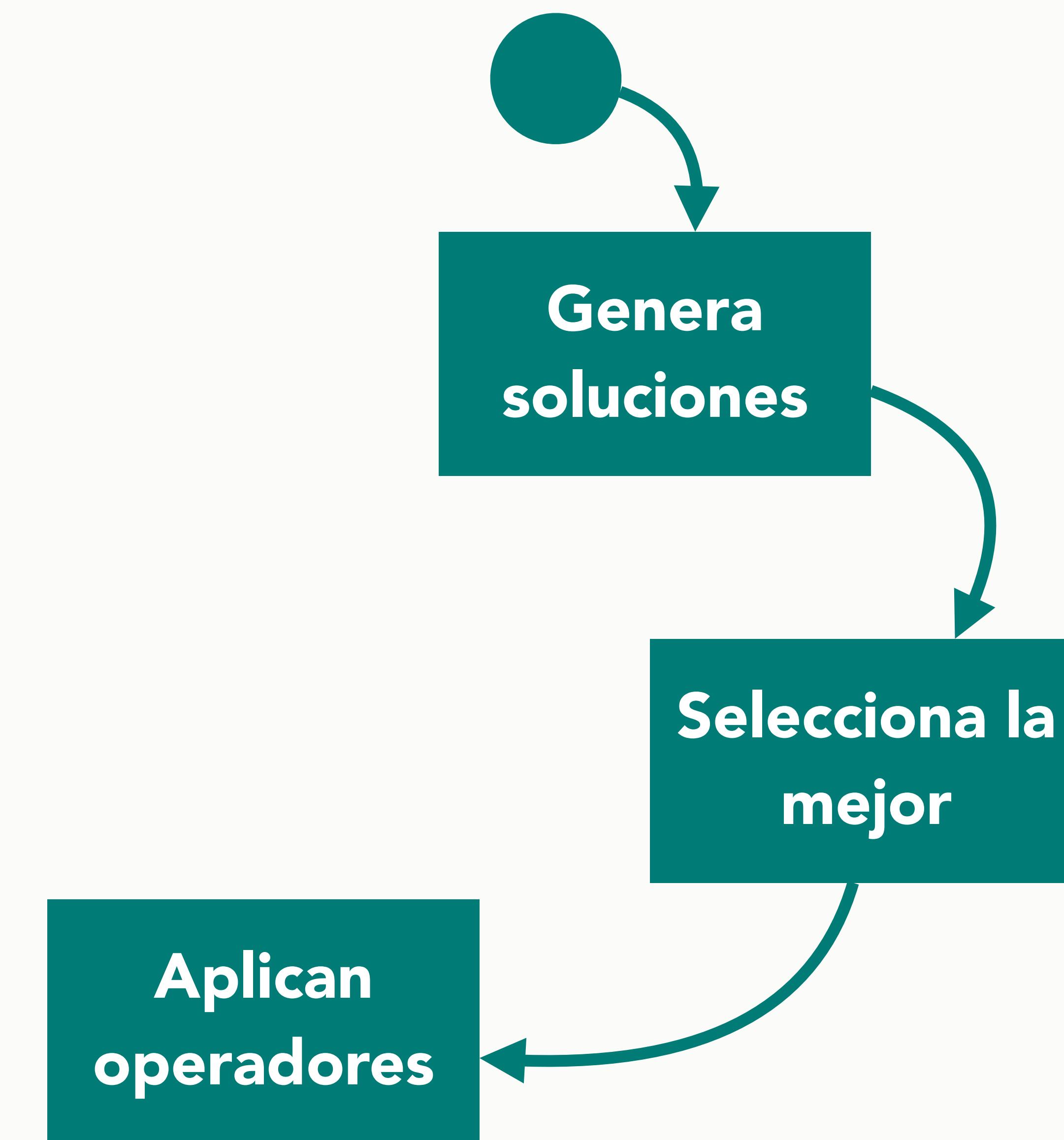
Estructura de la búsqueda tabú

fundamentos



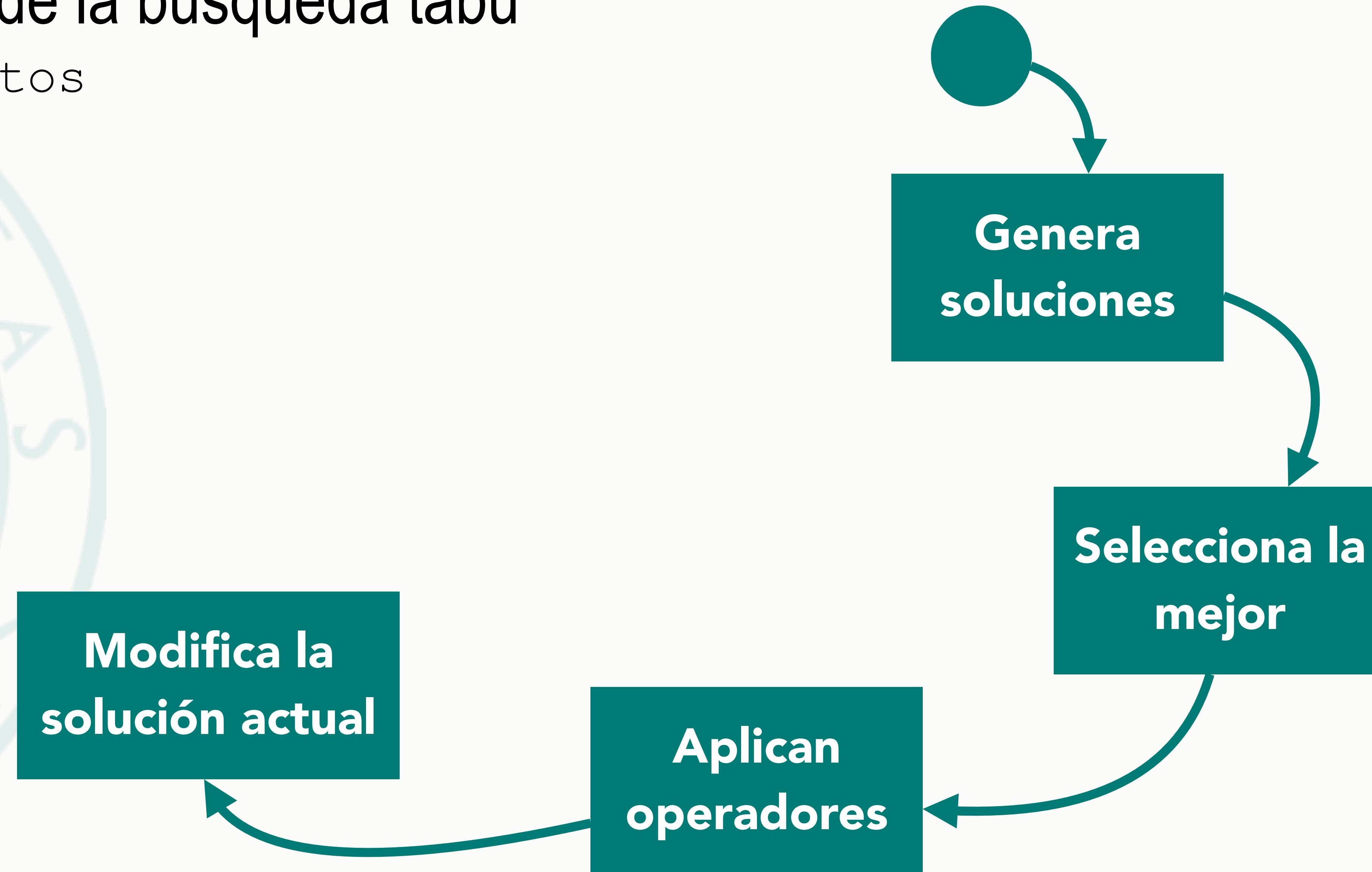
Estructura de la búsqueda tabú

fundamentos



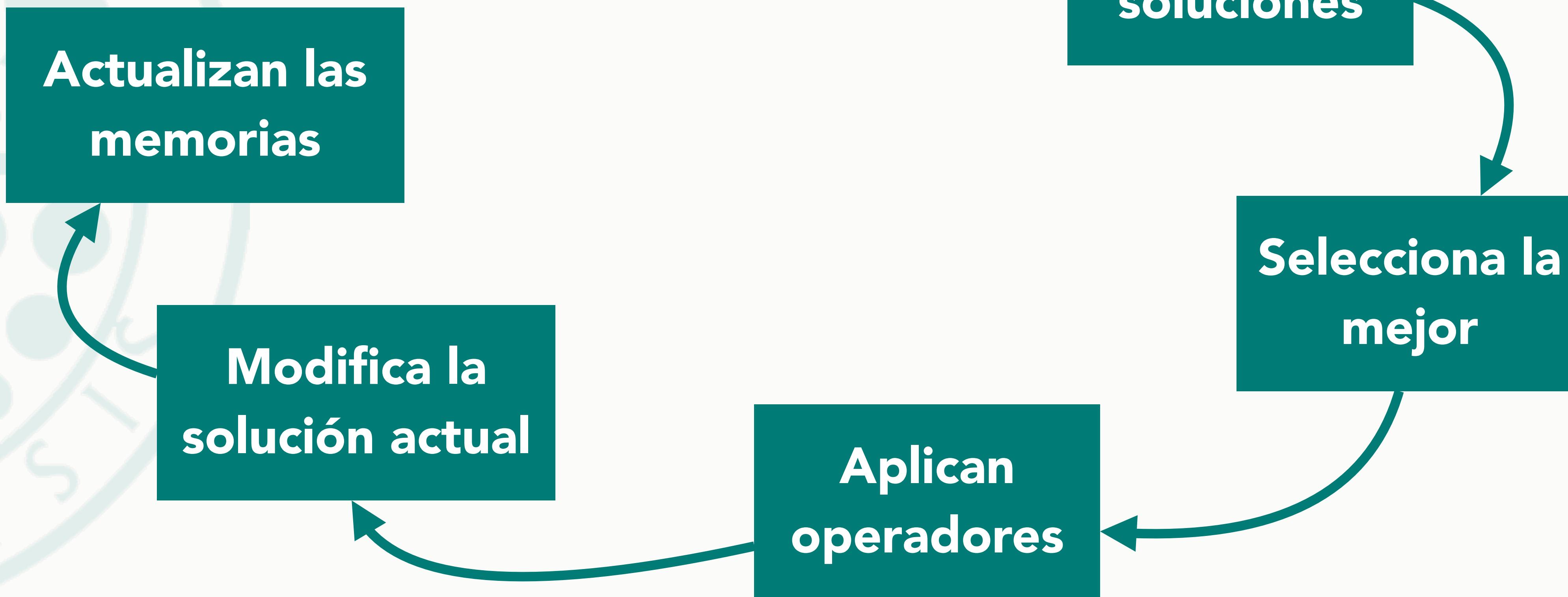
Estructura de la búsqueda tabú

fundamentos



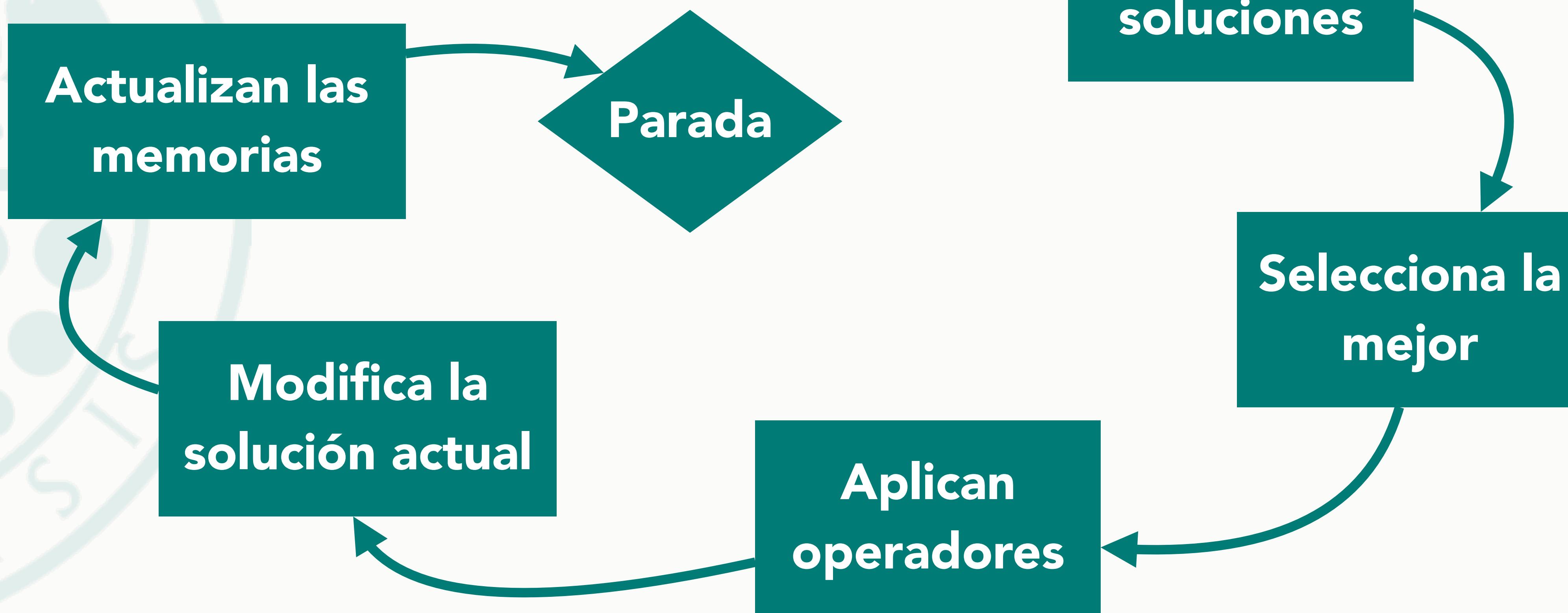
Estructura de la búsqueda tabú

fundamentos



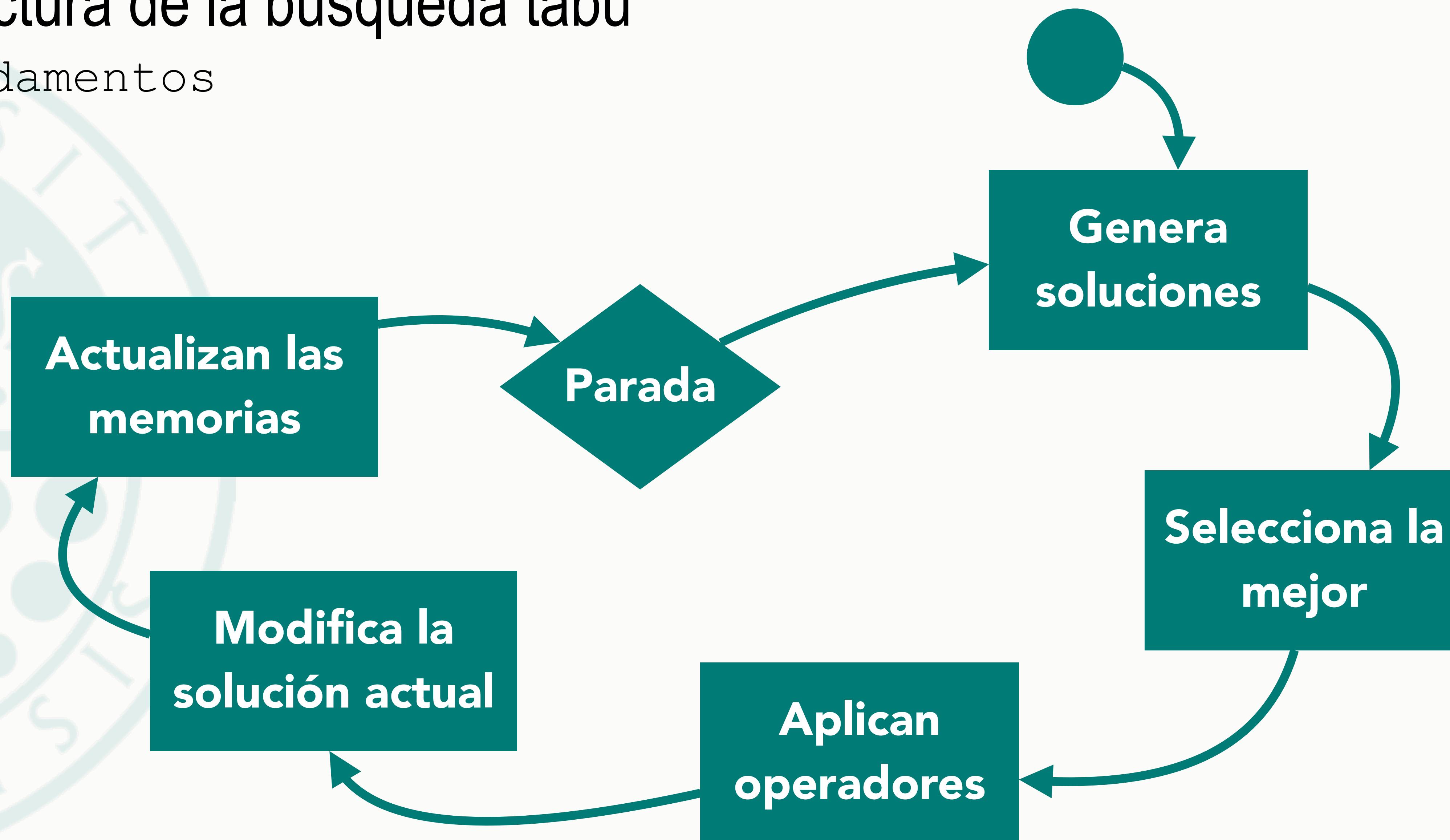
Estructura de la búsqueda tabú

fundamentos



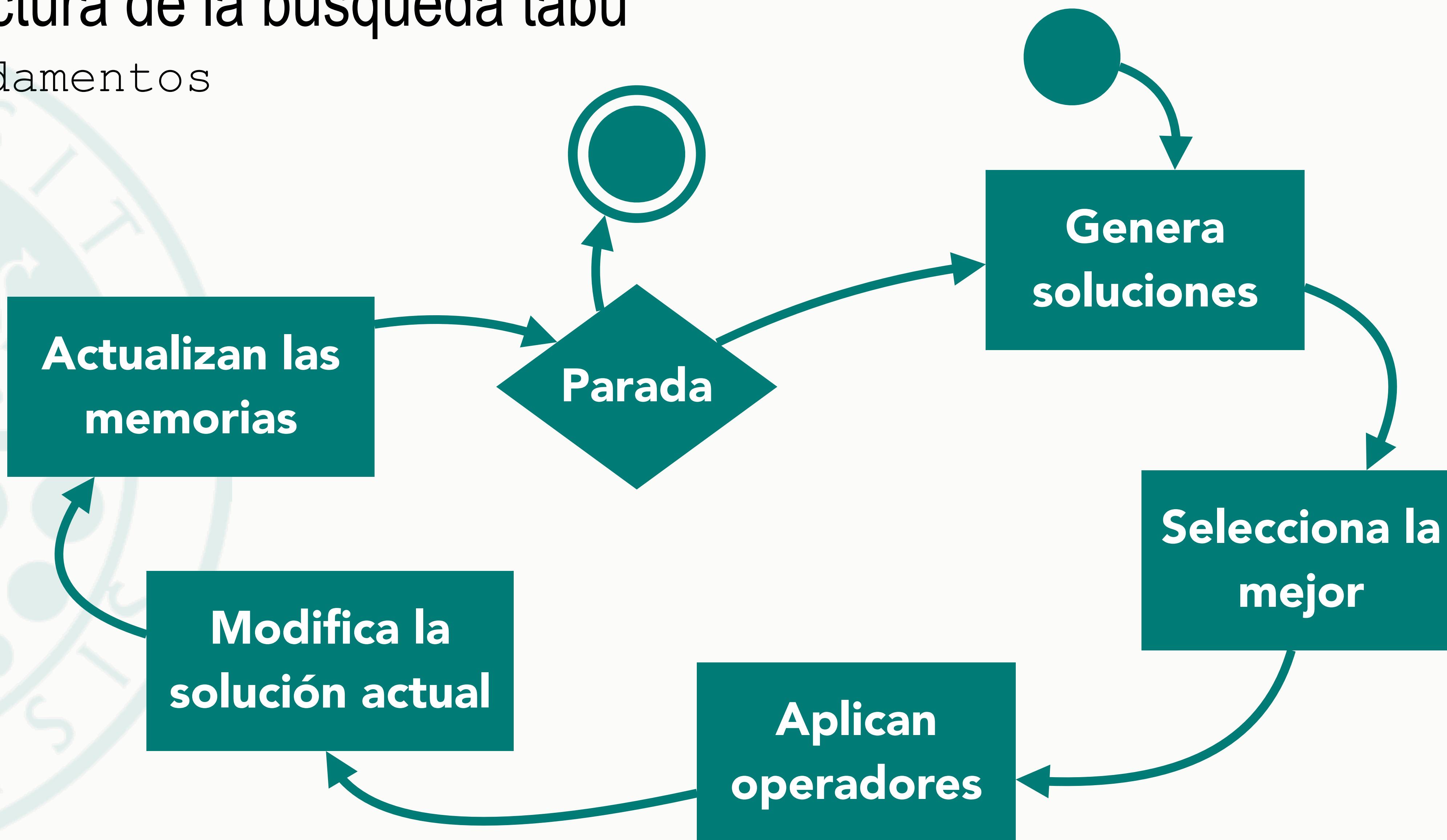
Estructura de la búsqueda tabú

fundamentos



Estructura de la búsqueda tabú

fundamentos



Estructura de la búsqueda tabú

fundamentos

Las estructuras de las memorias operan en 4 dimensiones



Estructura de la búsqueda tabú

fundamentos

Las estructuras de las memorias operan en 4 dimensiones



Estructura de la búsqueda tabú

fundamentos

Las estructuras de las memorias operan en 4 dimensiones



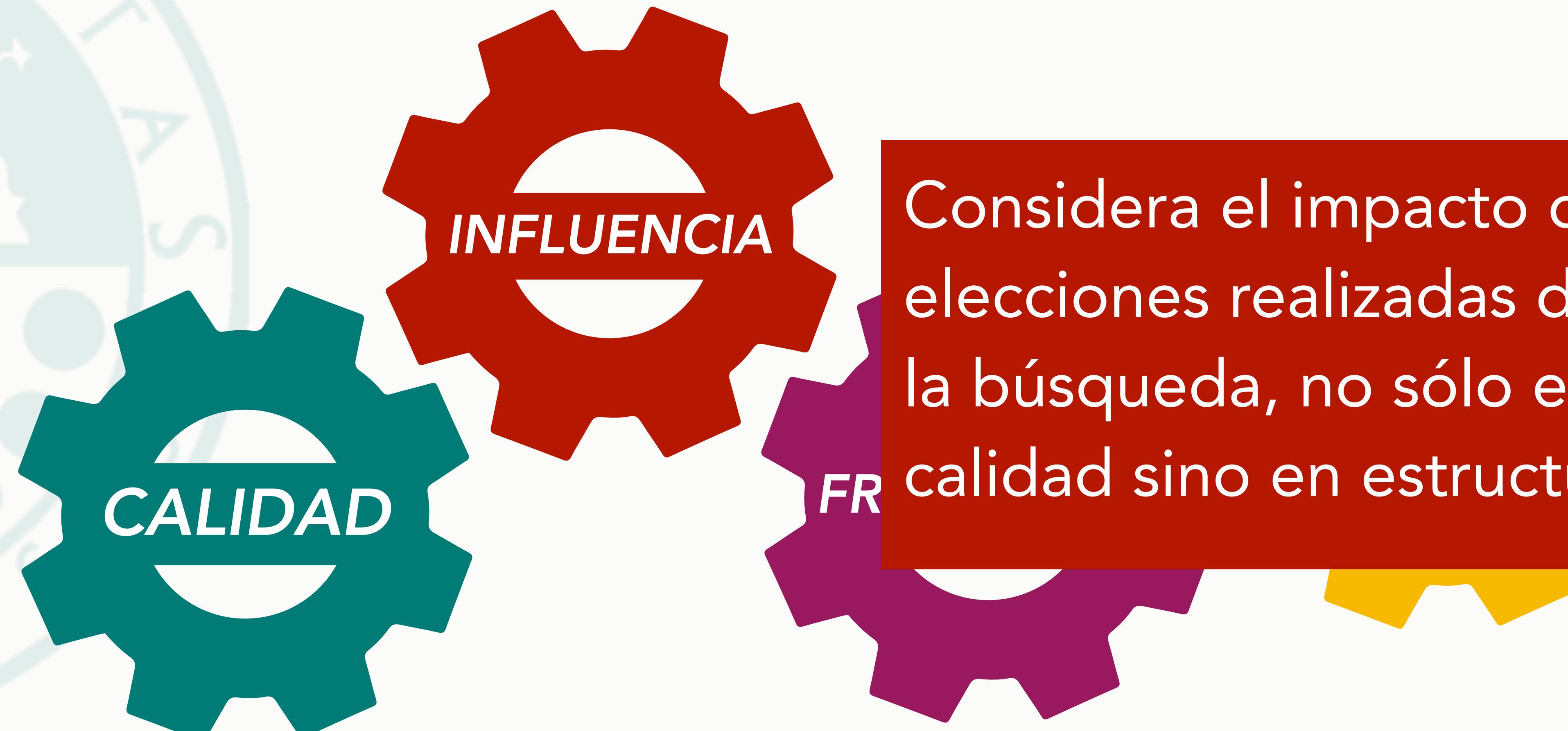
Diferencia el beneficio de las soluciones que se han visitado durante la búsqueda

Por ejemplo la identificación de elementos que son comunes a buenas soluciones

Estructura de la búsqueda tabú

fundamentos

Las estructuras de las memorias operan en 4 dimensiones



Estructura de la búsqueda tabú

fundamentos

La memoria en la búsqueda tabú puede ser:

- **Explícita** donde se almacenen soluciones completas
- **Implícita** recordando partes de la solución como por ejemplo movimientos o atributos, entre otros.

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- Extienda la búsqueda local sustituyendo el entorno actual por otro entorno, tal que, $E^*(S_{act}) \subset E(S_{act})$
- 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 a corto plazo **LISTA TABÚ (LT)** permite determinar este E^* y así organizar la manera en la que se explora el espacio

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- La LT es una memoria basada en movimientos recientes
- Las soluciones admitidas en E^* dependen de la estructura de la LT:

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- La LT es una memoria basada en movimientos recientes
- Las soluciones admitidas en E^* dependen de la estructura de la LT:

1. Lista de soluciones tabú

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- La LT es una memoria basada en movimientos recientes
- Las soluciones admitidas en E^* dependen de la estructura de la LT:

1. Lista de soluciones tabú

2. Lista de movimientos tabú

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- La LT es una memoria basada en movimientos recientes
- Las soluciones admitidas en E^* dependen de la estructura de la LT:

1. Lista de soluciones tabú

2. Lista de movimientos tabú

3. Lista de valores de atributos

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- La LT es una memoria basada en movimientos recientes
- Las soluciones admitidas en E^* dependen de la estructura de la LT:

1. Lista de soluciones tabú

2. Lista de movimientos tabú

3. Lista de valores de atributos

Se identifica soluciones ya visitadas y se marcan como tabú para no volver a ellas, eliminándolas del vecindario de S_{act}

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- La LT es una memoria basada en movimientos recientes
- Las soluciones admitidas en E^* dependen de la estructura de la LT:

1. Lista de soluciones tabú

2. Lista de movimientos tabú

3. Lista de valores de atributos

Se elimina del entorno todos los vecinos resultantes de aplicar sobre S_{act} un movimiento realizado anteriormente

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- La LT es una memoria basada en movimientos recientes
- Las soluciones admitidas en E^* dependen de la estructura de la LT:

1. Lista de soluciones tabú

2. Lista de movimientos tabú

3. Lista de valores de atributos

Se eliminan del entorno todos los vecinos con un par (atributo,valor) que presentara alguna solución explorada anteriormente

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

- La LT es una memoria basada en movimientos recientes
- Las soluciones admitidas en E^* dependen de la estructura de la LT:

Atributos o movimientos seleccionados son designados como “tabú-activos” y las soluciones que contienen elementos tabú-activos se convierten en “soluciones tabú”

Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

$L_{Tsol} = \{(1,3,2,4), (3,1,2,4), (3,2,1,4)\}$ || Operador de vecino: 2-opt

Solución = (1,2,3,4)

$E^*(S) = \{(2,1,3,4), (3,2,1,4), (4,2,3,1), (1,3,2,4), (1,4,3,2), (1,2,4,3)\}$

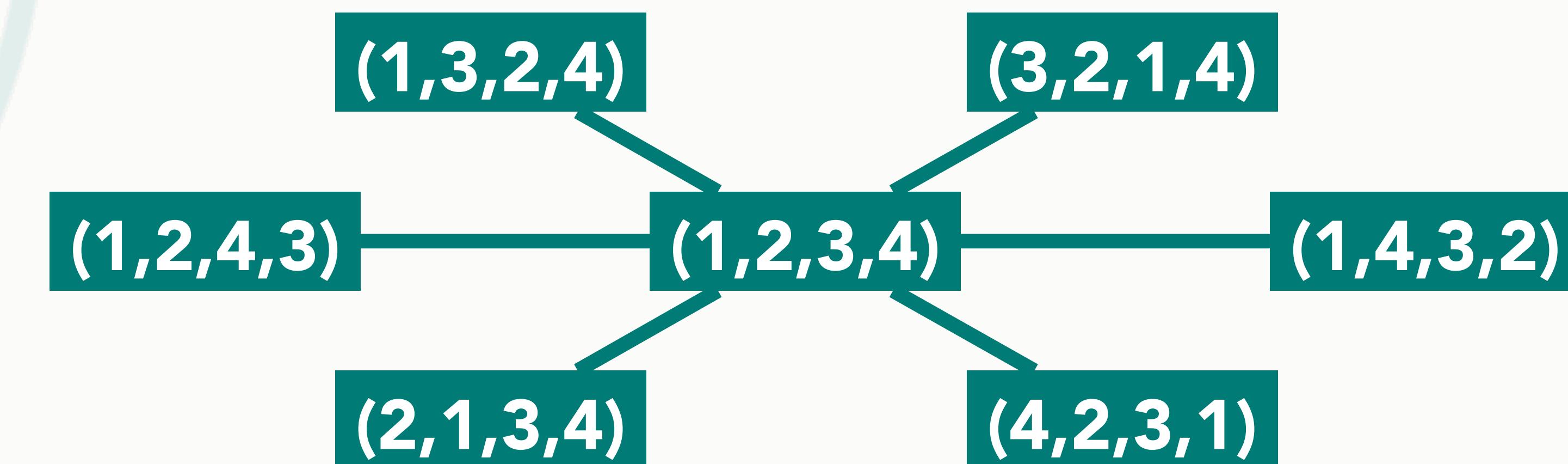
Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

$L_{Tsol} = \{(1,3,2,4), (3,1,2,4), (3,2,1,4)\}$ || Operador de vecino: 2-opt

Solución = (1,2,3,4)

$E^*(S) = \{(2,1,3,4), (3,2,1,4), (4,2,3,1), (1,3,2,4), (1,4,3,2), (1,2,4,3)\}$



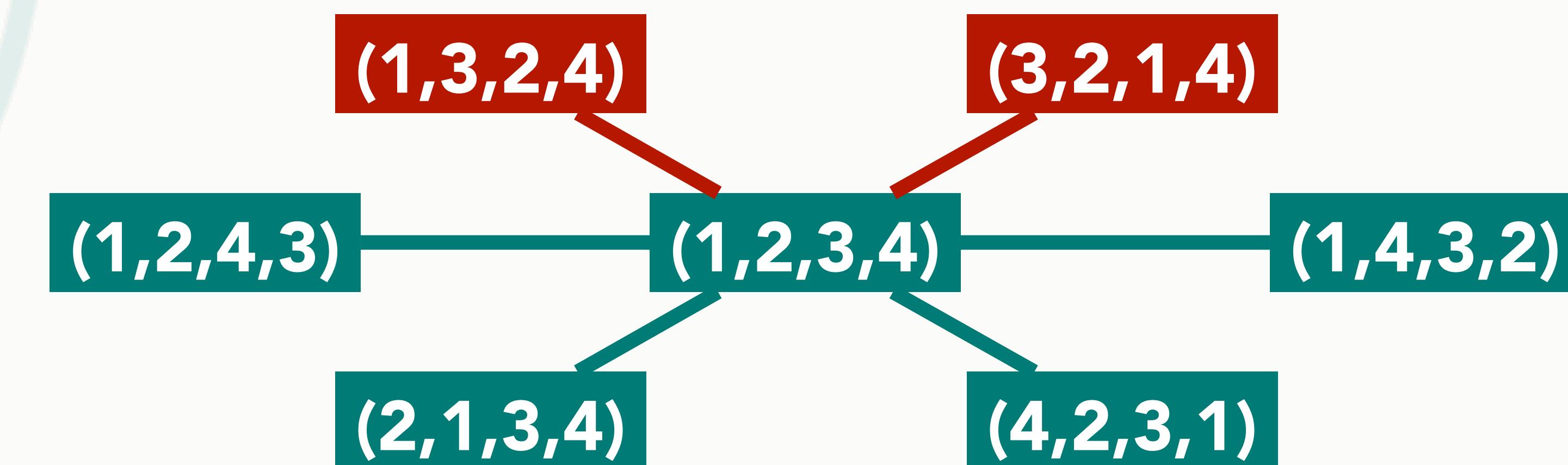
Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

$L_{Tsol} = \{(1,3,2,4), (3,1,2,4), (3,2,1,4)\}$ || Operador de vecino: 2-opt

Solución = (1,2,3,4)

$E^*(S) = \{(2,1,3,4), (3,2,1,4), (4,2,3,1), (1,3,2,4), (1,4,3,2), (1,2,4,3)\}$



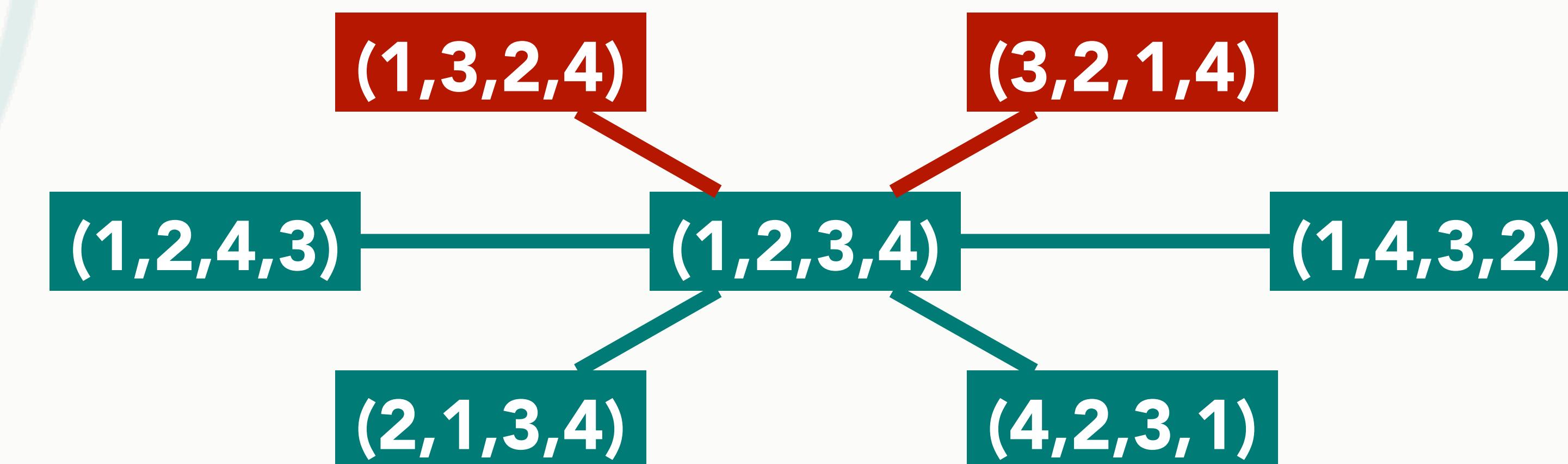
Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

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

Solución = $(1,2,3,4)$

$E^*(S) = \{(2,1,3,4), (3,2,1,4), (4,2,3,1), (1,3,2,4), (1,4,3,2), (1,2,4,3)\}$



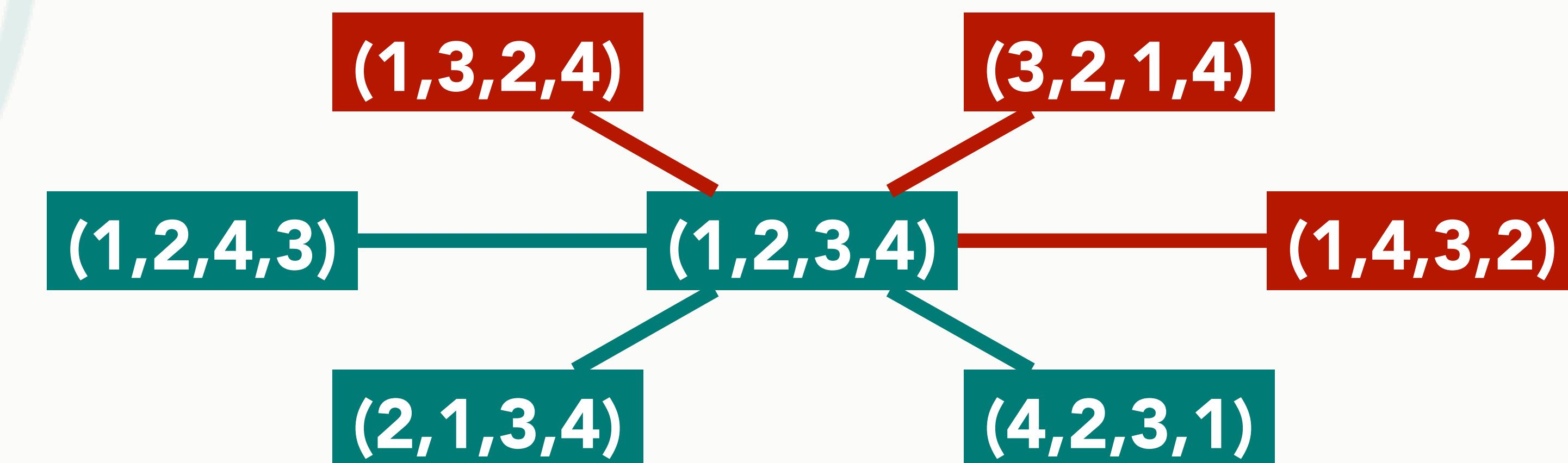
Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

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

Solución = $(1,2,3,4)$

$E^*(S) = \{(2,1,3,4), (3,2,1,4), (4,2,3,1), (1,3,2,4), (1,4,3,2), (1,2,4,3)\}$



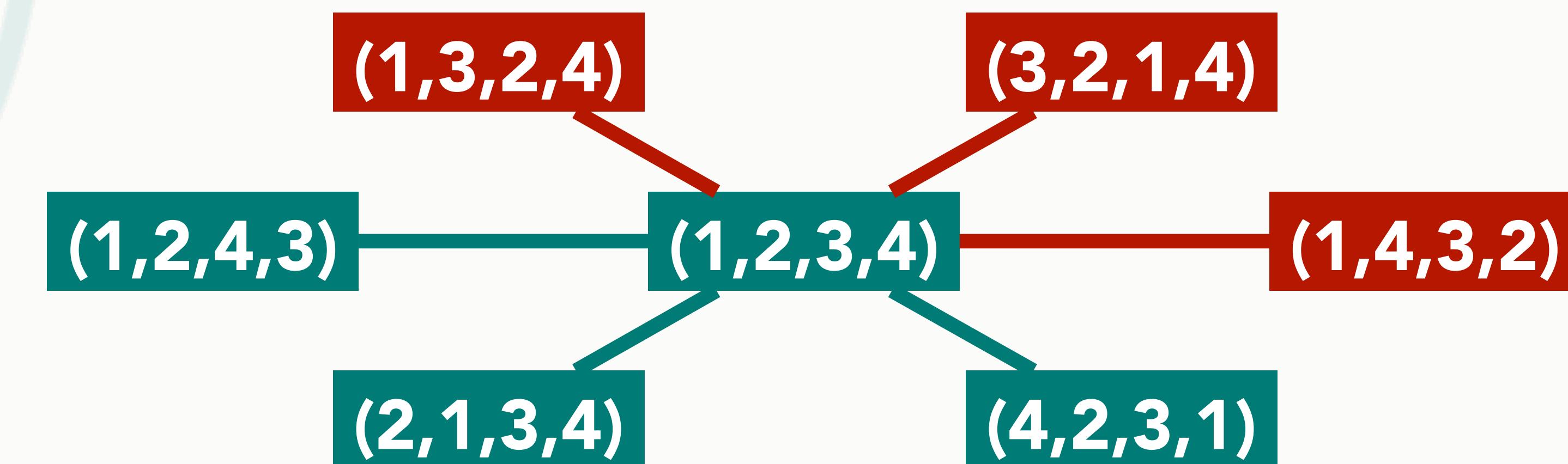
Estructura de la búsqueda tabú

memoria de corto plazo - generación entornos restringidos

LTmovimientos = {(1,3), **(2,4)**} || Operador de vecino: 2-opt

Solución = **(1,2,3,4)**

$E^*(S) = \{(2,1,3,4), (3,2,1,4), (4,2,3,1), (1,3,2,4), (1,4,3,2), (1,2,4,3)\}$



Estructura de la búsqueda tabú

memoria de corto plazo – generación entornos restringidos
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

Estructura de la búsqueda tabú

memoria de corto plazo – generación entornos restringidos

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

Estructura de la búsqueda tabú

memoria de corto plazo – generación entornos restringidos

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

Estructura de la búsqueda tabú

memoria de corto plazo – generación entornos restringidos

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 $\langle i, j \rangle$

Estructura de la búsqueda tabú

memoria de corto plazo – generación entornos restringidos

estructuras de memoria de corto plazo

- 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., $[n/3, 3 \cdot n]$)

Estructura de la búsqueda tabú



Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva
solución candidata S' a partir de S_{act}

Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva solución candidata S' a partir de S_{act}



Prueba Tabú

Comprobar si el movimiento para obtener S' es tabú-activo

Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva solución candidata S' a partir de S_{act}

Prueba Tabú

Comprobar si el movimiento para obtener S' es tabú-activo

NO

Evaluar S'

$$C_{vecino} = C(S')$$

Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva solución candidata S' a partir de S_{act}

Actualización del Mejor Vecino

$$C_{mejor_vecino} = C_{vecino} \text{ AND}$$

$$S_{mejor_vecino} = S'$$

Prueba Tabú

Comprobar si el movimiento para obtener S' es tabú-activo

NO

Evaluar S'

$$C_{vecino} = C(S')$$

Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva solución candidata S' a partir de S_{act}

Actualización del Mejor Vecino

$$C_{mejor_vecino} = C_{vecino} \text{ AND}$$

$$S_{mejor_vecino} = S'$$

Prueba Tabú

Comprobar si el movimiento para obtener S' es tabú-activo

NO

Evaluar S'

$$C_{vecino} = C(S')$$

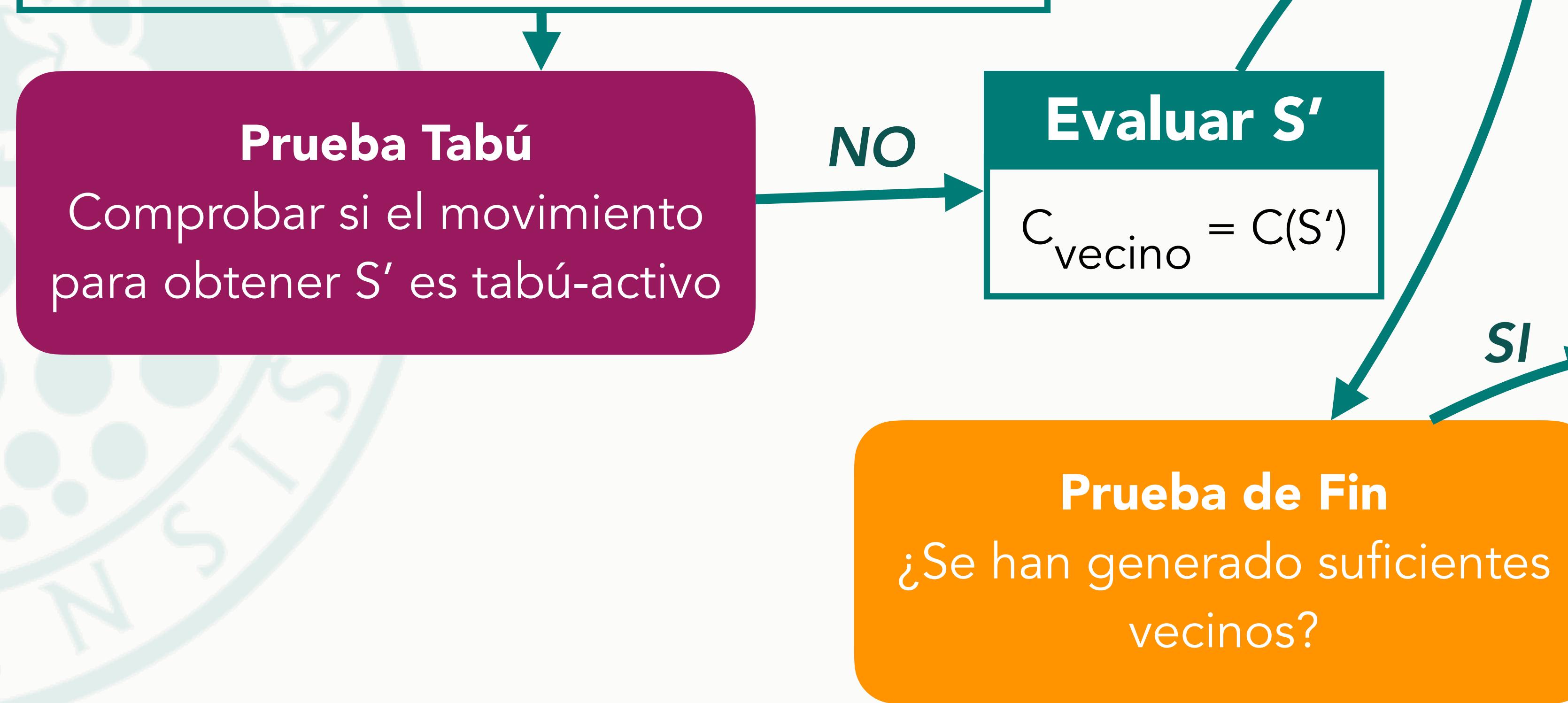
Prueba de Fin

¿Se han generado suficientes vecinos?

Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva solución candidata S' a partir de S_{act}



Actualización del Mejor Vecino

$$C_{mejor_vecino} = C_{vecino} \text{ AND}$$

$$S_{mejor_vecino} = S'$$

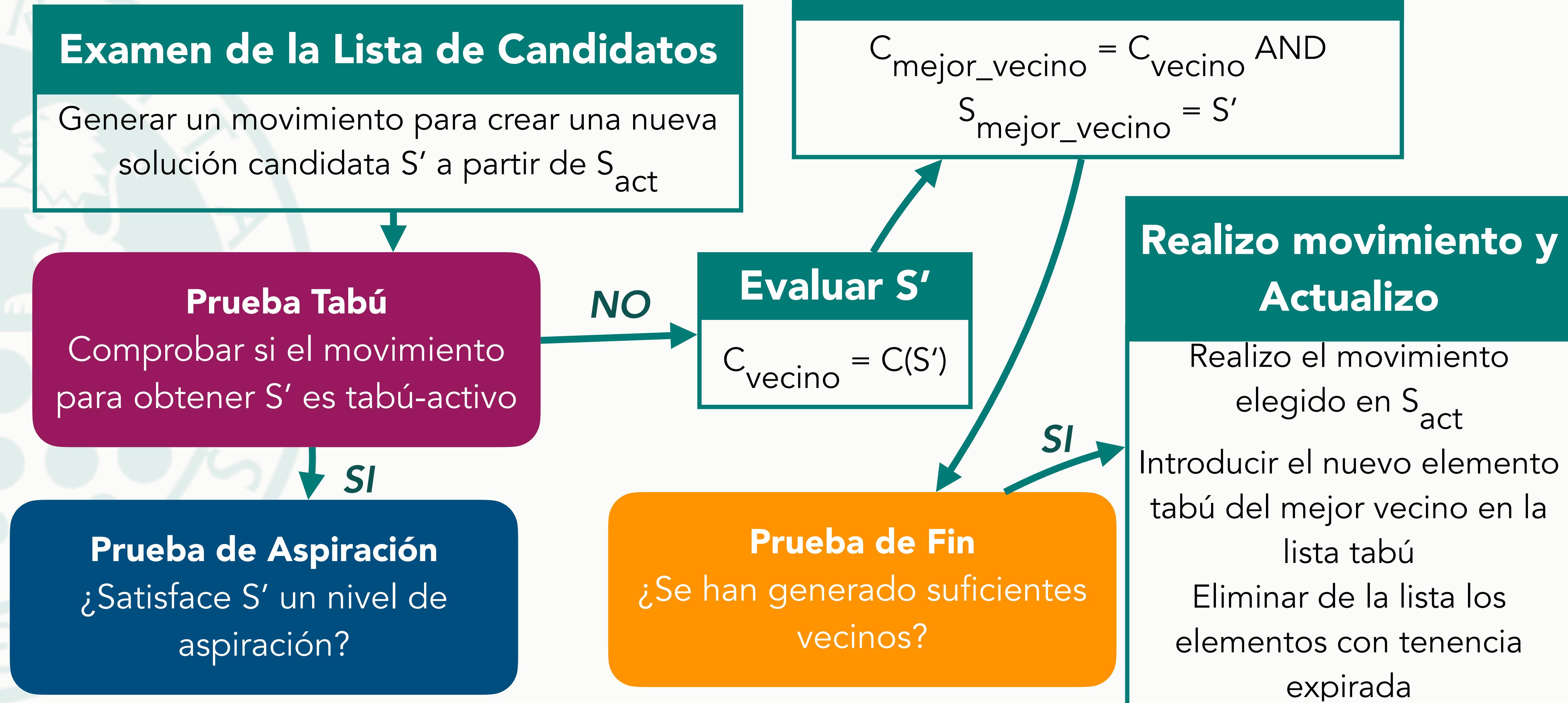
Realizo movimiento y Actualizo

Realizo el movimiento elegido en S_{act}

Introducir el nuevo elemento tabú del mejor vecino en la lista tabú

Eliminar de la lista los elementos con tenencia expirada

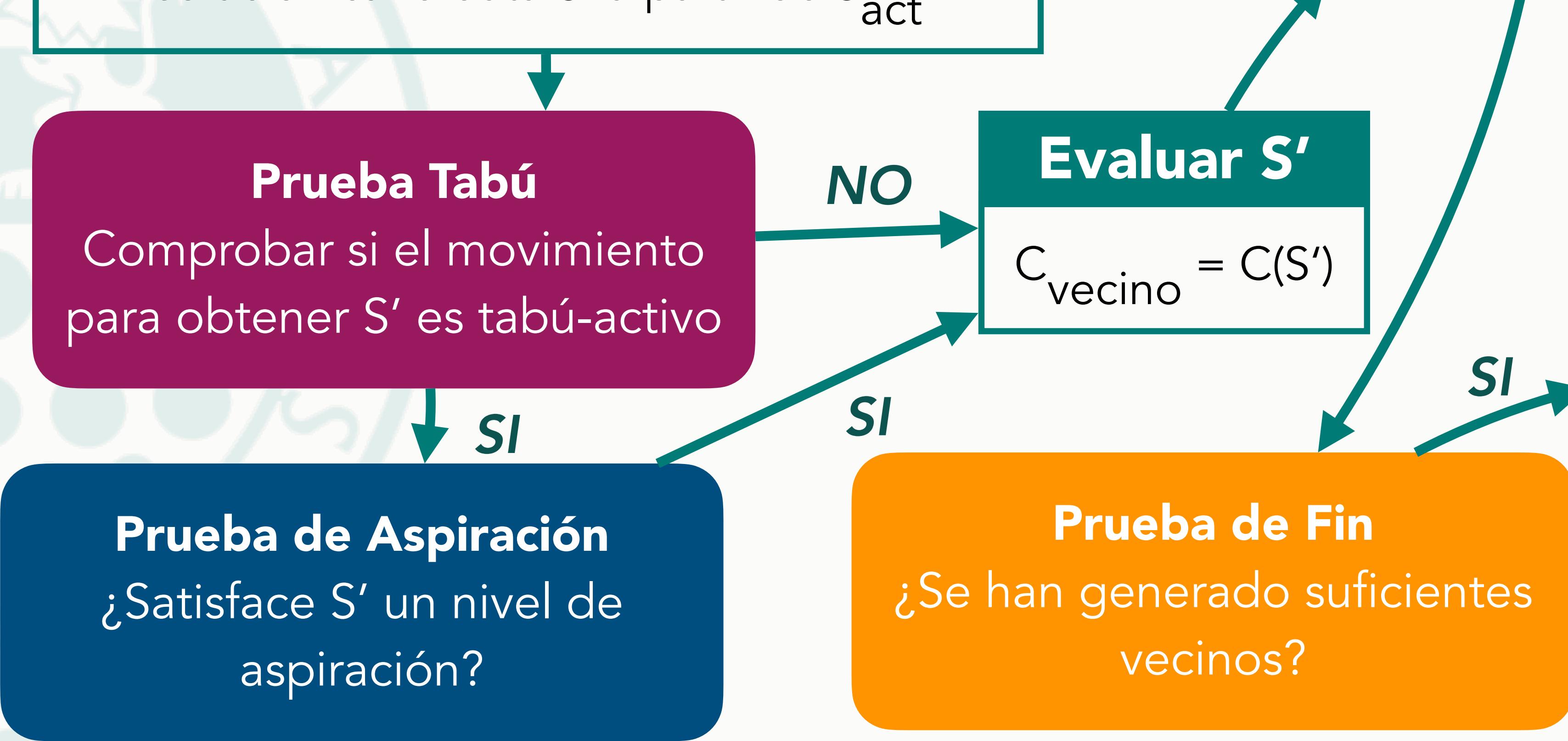
Estructura de la búsqueda tabú



Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva solución candidata S' a partir de S_{act}



Actualización del Mejor Vecino

$$C_{mejor_vecino} = C_{vecino} \text{ AND } S_{mejor_vecino} = S'$$

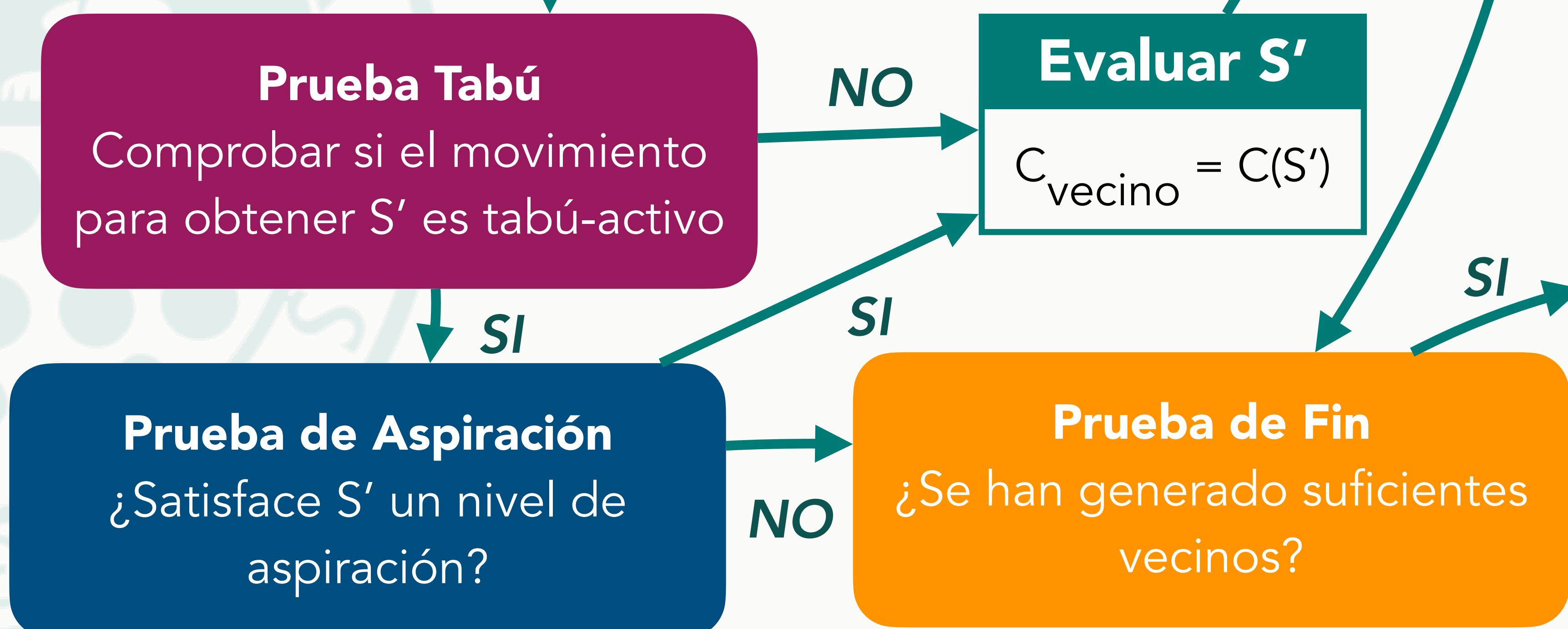
Realizo movimiento y Actualizo

Realizo el movimiento elegido en S_{act}
 Introducir el nuevo elemento tabú del mejor vecino en la lista tabú
 Eliminar de la lista los elementos con tenencia expirada

Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva solución candidata S' a partir de S_{act}



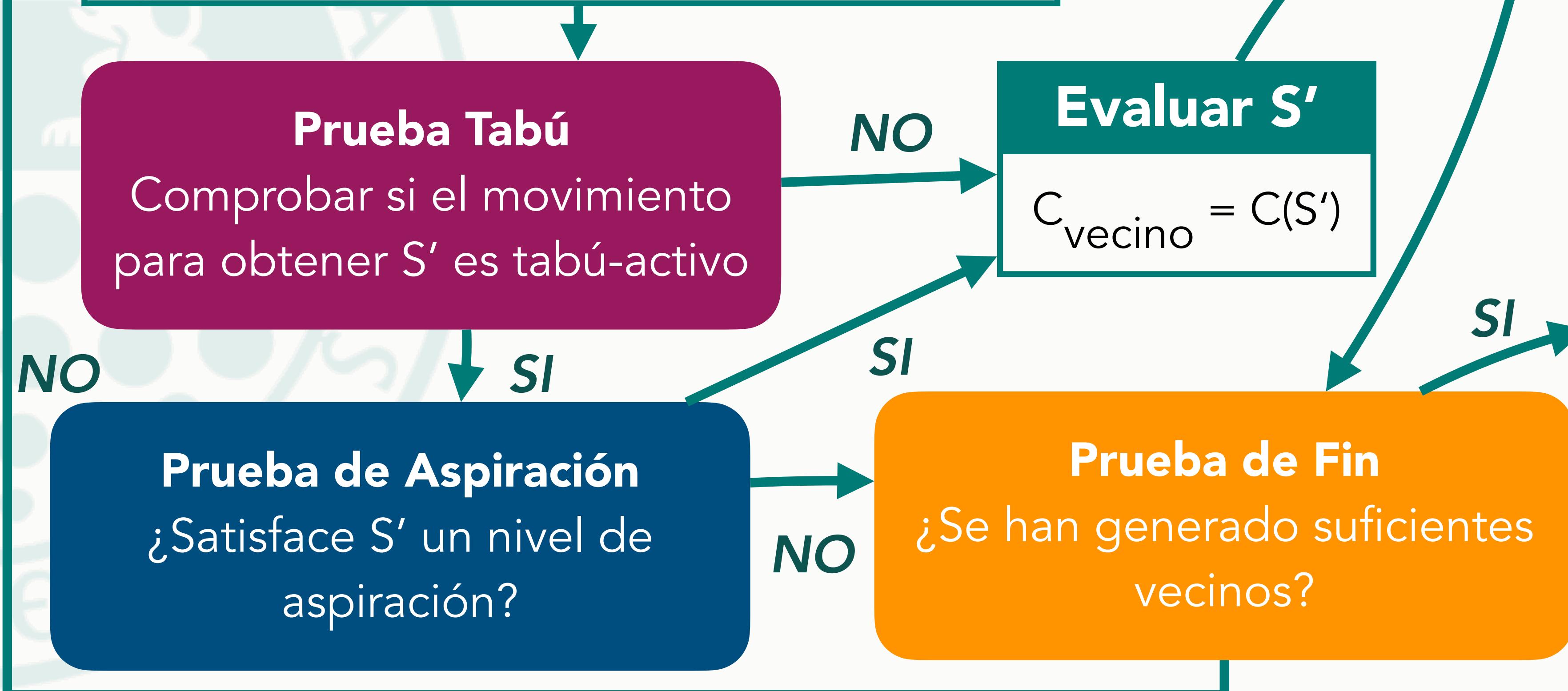
Estructura de la búsqueda tabú

Examen de la Lista de Candidatos

Generar un movimiento para crear una nueva solución candidata S' a partir de S_{act}

Actualización del Mejor Vecino

$$C_{mejor_vecino} = C_{vecino} \text{ AND } S_{mejor_vecino} = S'$$



Realizo movimiento y Actualizo

Realizo el movimiento elegido en S_{act}
 Introducir el nuevo elemento tabú del mejor vecino en la lista tabú
 Eliminar de la lista los elementos con tenencia expirada

Estructura de la búsqueda tabú

memoria de largo plazo - intensificación y diversificación

- 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

Estructura de la búsqueda tabú

memoria de largo plazo

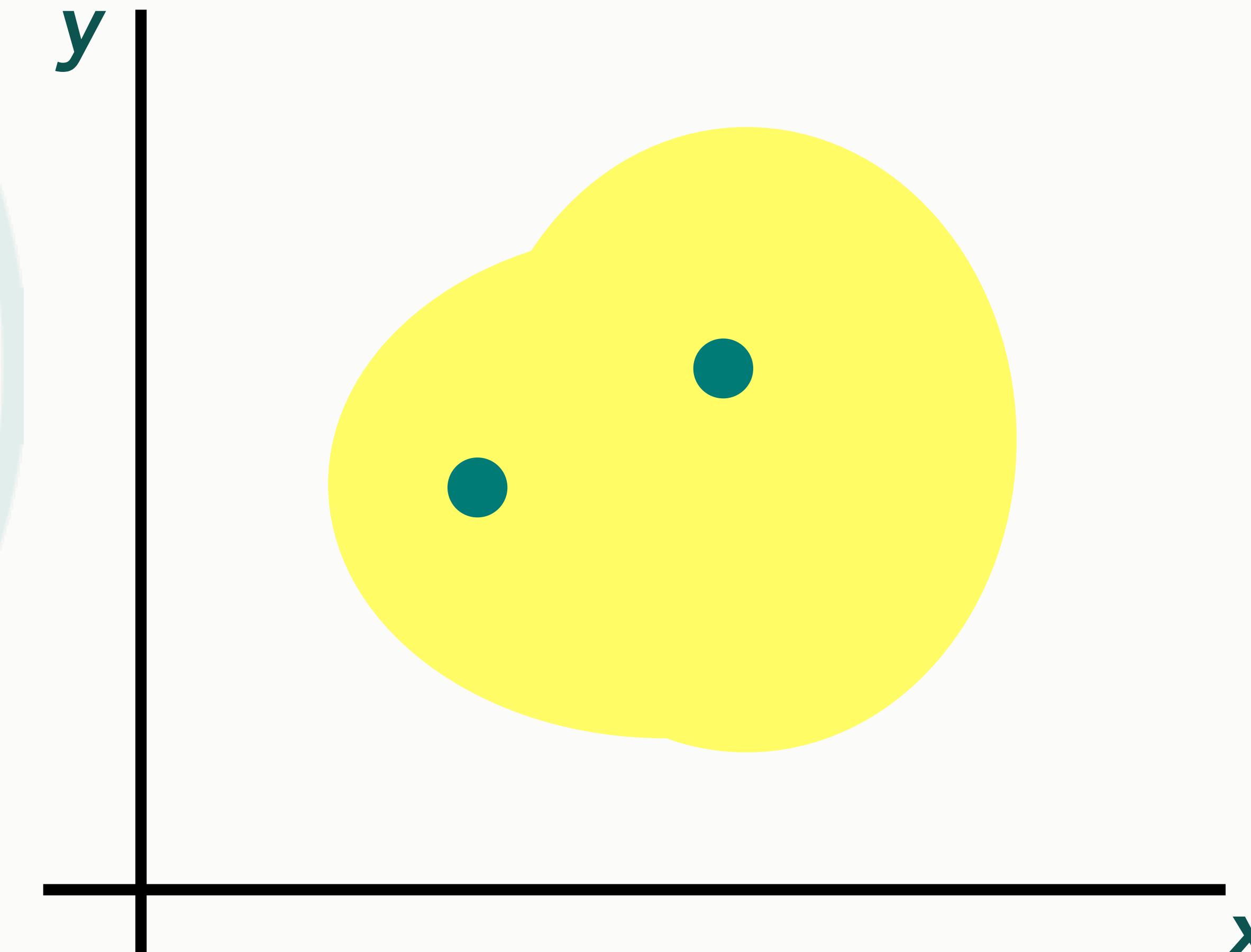
intensificación

- 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

Estructura de la búsqueda tabú

memoria de largo plazo

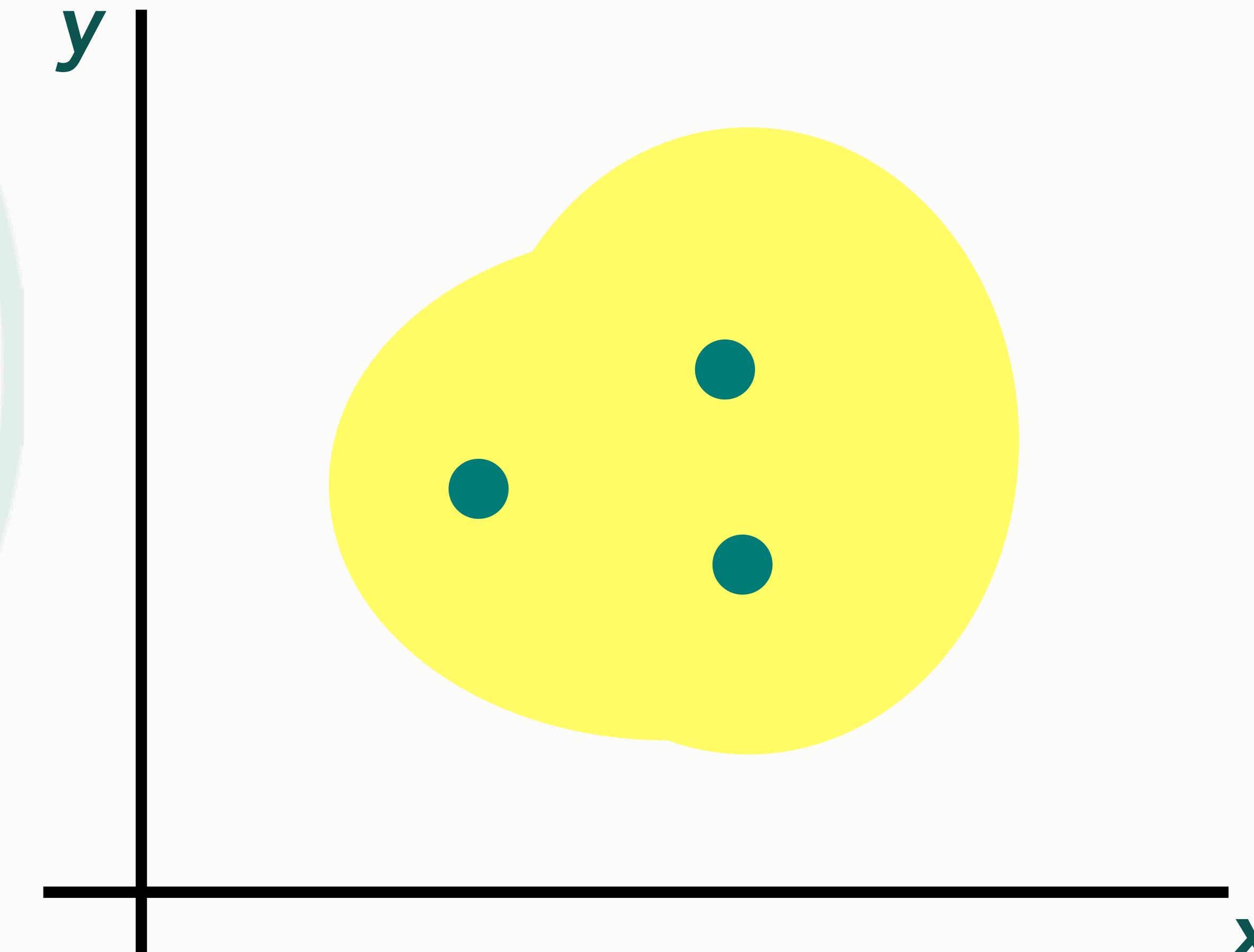
intensificación



Estructura de la búsqueda tabú

memoria de largo plazo

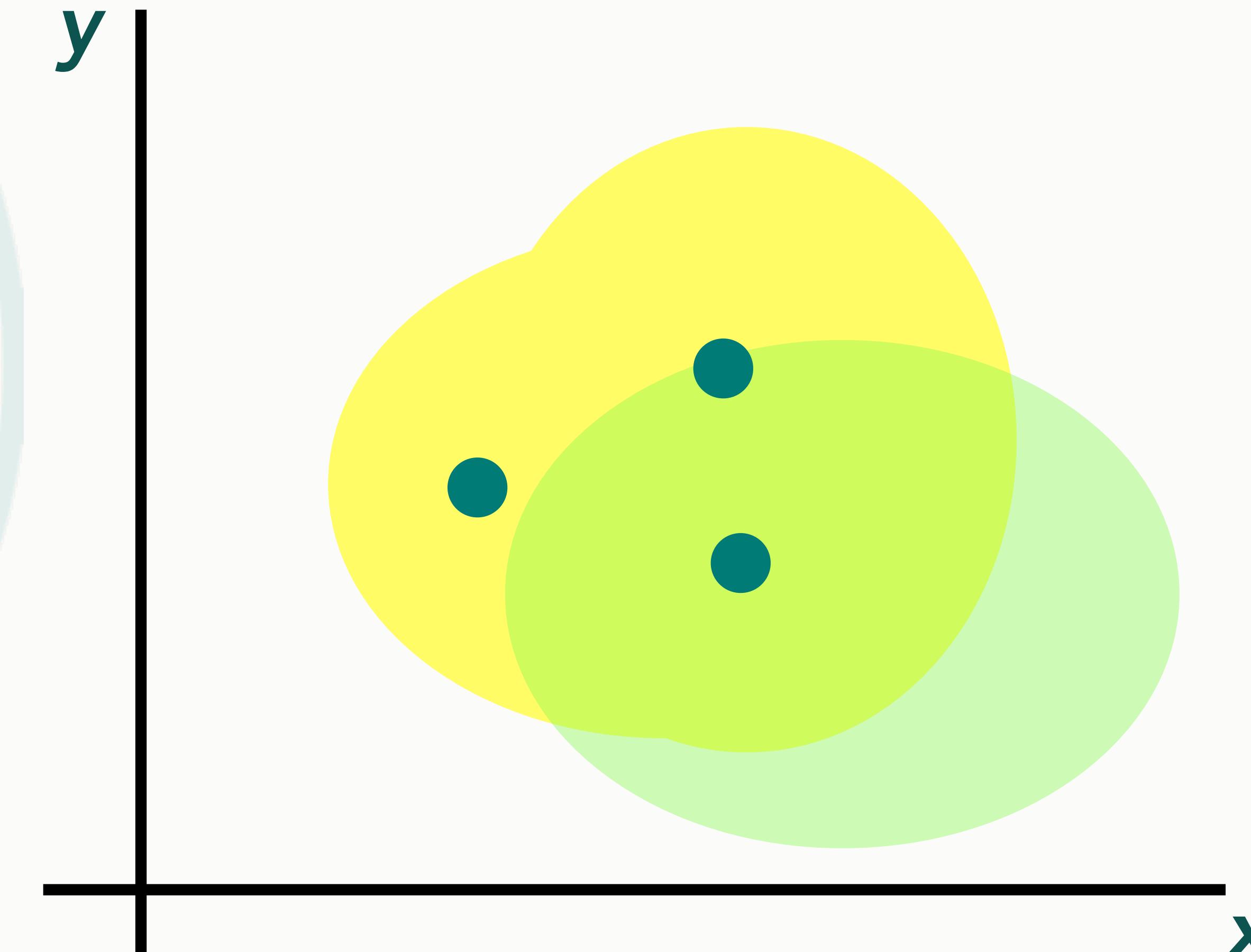
intensificación



Estructura de la búsqueda tabú

memoria de largo plazo

intensificación



Estructura de la búsqueda tabú

memoria de largo plazo

intensificación

- 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

Estructura de la búsqueda tabú

memoria de largo plazo

intensificación



Estructura de la búsqueda tabú

memoria de largo plazo

intensificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo



Estructura de la búsqueda tabú

memoria de largo plazo

intensificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Crear e ir manteniendo una lista de k soluciones élite ($k=5$ ó 20)

Estructura de la búsqueda tabú

memoria de largo plazo

intensificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Crear e ir manteniendo una lista de k soluciones élite ($k=5$ ó 20)

Cuando el algoritmo se estanque (es decir, cuando la proporción de aceptación de nuevas soluciones mejores caiga por debajo de un umbral):

Si se llegó al límite de iteraciones o la lista de soluciones élite está vacía **Parar**

Si no, escoger una solución élite y eliminarla de la lista

Estructura de la búsqueda tabú

memoria de largo plazo

intensificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Crear e ir manteniendo una lista de k soluciones élite ($k=5$ ó 20)

Cuando el algoritmo se estanque (es decir, cuando la proporción de aceptación de nuevas soluciones mejores caiga por debajo de un umbral):

Si se llegó al límite de iteraciones o la lista de soluciones élite está vacía **Parar**

Si no, escoger una solución élite y eliminarla de la lista

Continuar la ejecución del algoritmo de TS desde la solución escogida (opcionalmente restaurar la memoria de corto plazo)

Si durante las iteraciones actuales se encuentran soluciones válidas para la estrategia de selección de soluciones élite, incluirlas en la lista (reemplazando a otras si lista llena)

Estructura de la búsqueda tabú

memoria de largo plazo

intensificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Crear e ir manteniendo una lista de k soluciones élite ($k=5$ ó 20)

Cuando el algoritmo se estanque (es decir, cuando la proporción de aceptación de nuevas soluciones mejores caiga por debajo de un umbral):

Si se llegó al límite de iteraciones o la lista de soluciones élite está vacía **Parar**

Si no, escoger una solución élite y eliminarla de la lista

Continuar la ejecución del algoritmo de TS desde la solución escogida (opcionalmente restaurar la memoria de corto plazo)

Si durante las iteraciones actuales se encuentran soluciones válidas para la estrategia de selección de soluciones élite, incluirlas en la lista (reemplazando a otras si lista llena)

Estructura de la búsqueda tabú

memoria de largo plazo

estrategias de diversificación

Conducen la búsqueda hacia nuevas regiones del espacio de búsqueda no exploradas aún

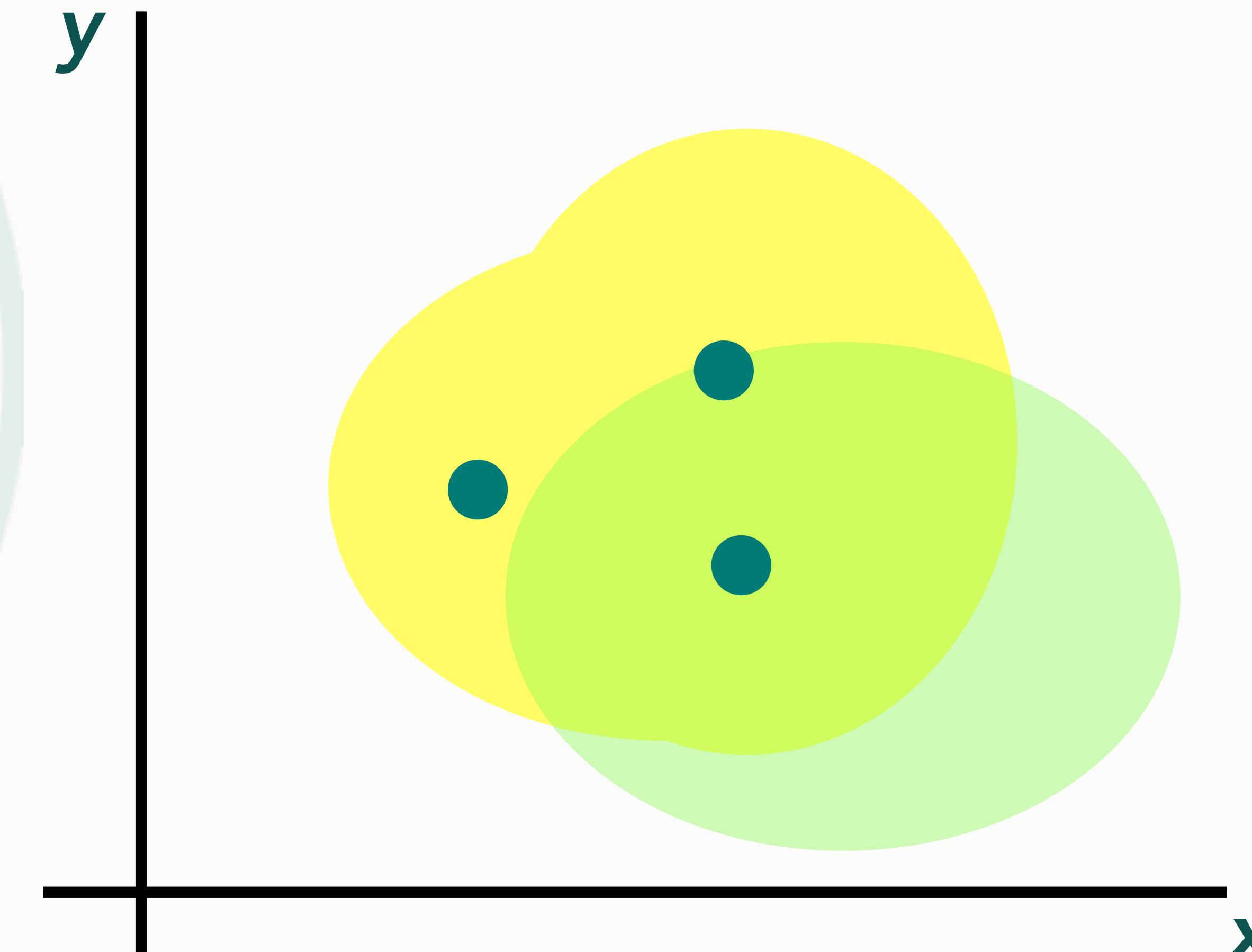
La búsqueda se reinicializa cuando se estanca, partiendo de una solución no visitada

Esta solución se genera a partir de la memoria de frecuencias, dando **mayor probabilidad de aparición a los valores menos habituales**

Estructura de la búsqueda tabú

memoria de largo plazo

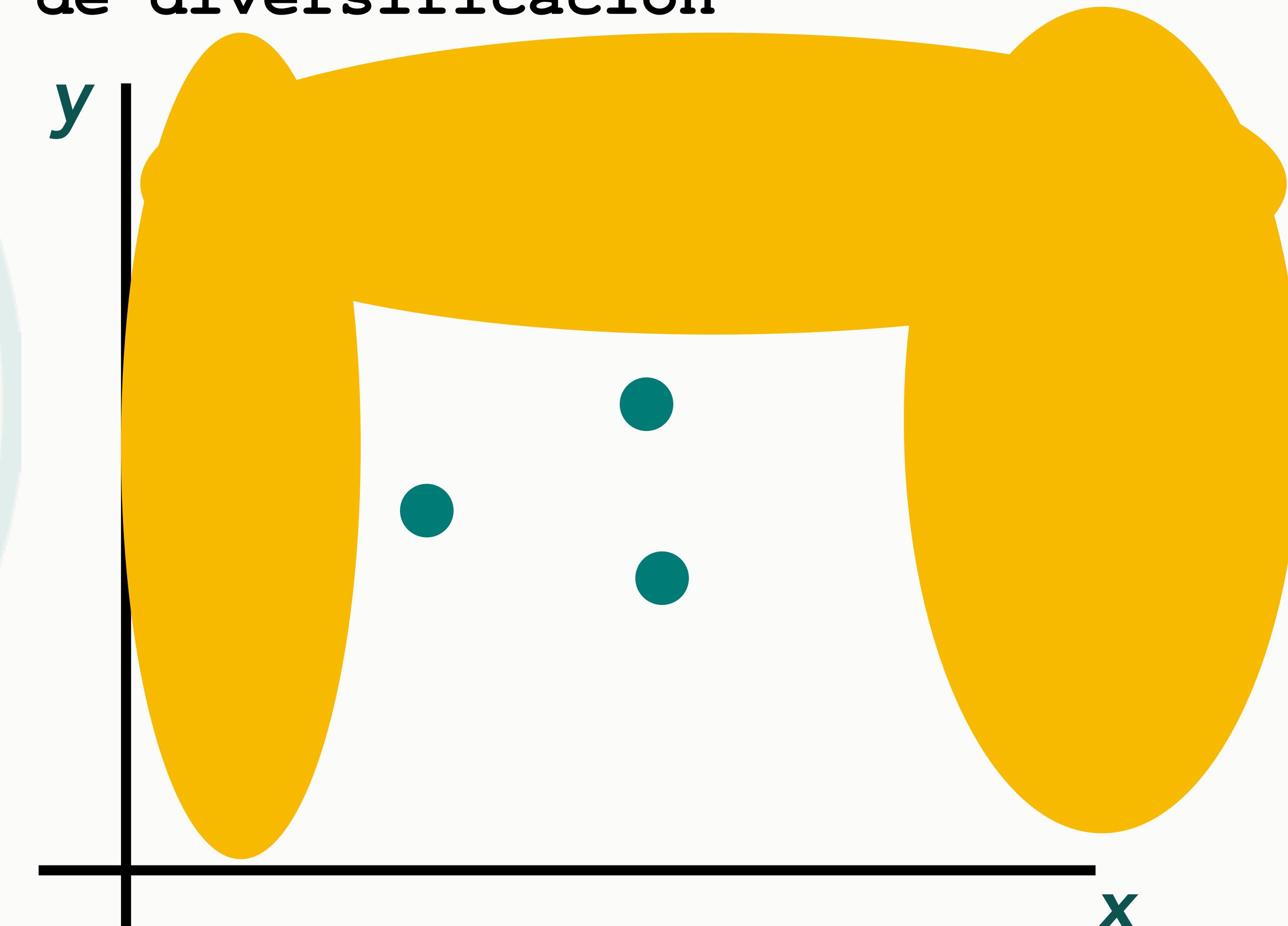
estrategias de diversificación



Estructura de la búsqueda tabú

memoria de largo plazo

estrategias de diversificación



Estructura de la búsqueda tabú

enfoque simple de diversificación



Estructura de la búsqueda tabú

enfoque simple de diversificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo



Estructura de la búsqueda tabú

enfoque simple de diversificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Mantener la memoria de frecuencias en función de los valores de atributos de las soluciones aceptadas

Estructura de la búsqueda tabú

enfoque simple de diversificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Cuando el algoritmo se estanke, generar aleatoriamente una nueva solución inicial a partir de la memoria de frecuencias, dando mayor probabilidad de aparición a los valores de atributos menos usados

Mantener la memoria de frecuencias en función de los valores de atributos de las soluciones aceptadas

Estructura de la búsqueda tabú

enfoque simple de diversificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Cuando el algoritmo se estanke, generar aleatoriamente una nueva solución inicial a partir de la memoria de frecuencias, dando mayor probabilidad de aparición a los valores de atributos menos usados

Mantener la memoria de frecuencias en función de los valores de atributos de las soluciones aceptadas

Continuar la búsqueda a partir de esa solución usando la memoria de corto plazo de la forma habitual
Seguir actualizando la memoria de frecuencias

Estructura de la búsqueda tabú

enfoque simple de diversificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Mantener la memoria de frecuencias en función de los valores de atributos de las soluciones aceptadas

Cuando el algoritmo se estanque, generar aleatoriamente una nueva solución inicial a partir de la memoria de frecuencias, dando mayor probabilidad de aparición a los valores de atributos menos usados

Continuar la búsqueda a partir de esa solución usando la memoria de corto plazo de la forma habitual
Seguir actualizando la memoria de frecuencias

Cuando el algoritmo se estanque de nuevo, estudiar si se da la condición de parada global.
Si es así, Parar, si no, generar una nueva solución inicial

Estructura de la búsqueda tabú

enfoque simple de diversificación

Realizar serie de iteraciones haciendo uso de la memoria de corto plazo

Mantener la memoria de frecuencias en función de los valores de atributos de las soluciones aceptadas

Cuando el algoritmo se estanque, generar aleatoriamente una nueva solución inicial a partir de la memoria de frecuencias, dando mayor probabilidad de aparición a los valores de atributos menos usados

Continuar la búsqueda a partir de esa solución usando la memoria de corto plazo de la forma habitual
Seguir actualizando la memoria de frecuencias

Cuando el algoritmo se estanque de nuevo, estudiar si se da la condición de parada global.
Si es así, Parar, si no, generar una nueva solución inicial

Estructura de la búsqueda tabú

memoria de largo plazo

oscilación estratégica

Interacción efectiva entre intensificación y diversificación

consistente en escoger aleatoriamente una u otra cuando la búsqueda se estanque

Estructura de la búsqueda tabú

memoria de largo plazo

estructuras de memoria de largo plazo

- 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 : $M[i,j]$

Estructura de la búsqueda tabú

memoria de largo plazo

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**

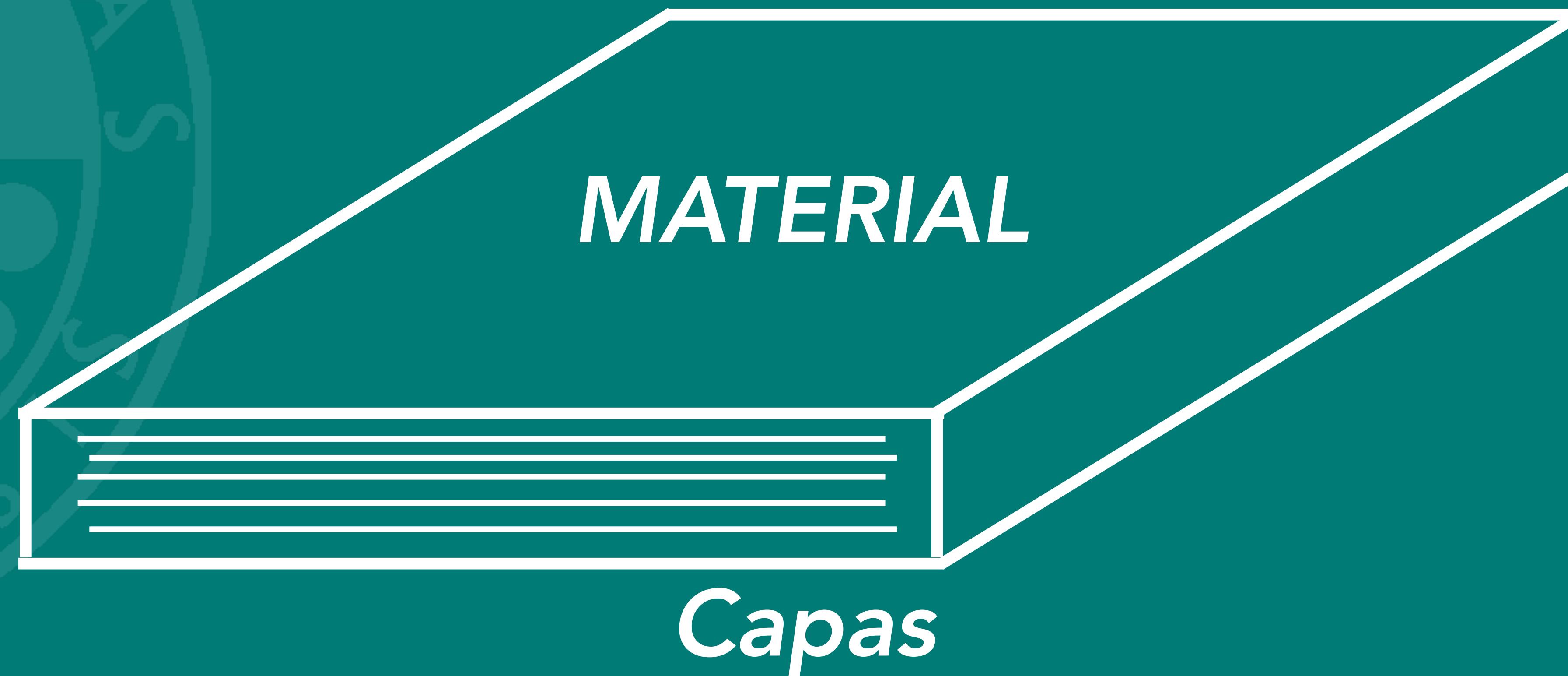
Ejemplo

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

Ejemplo

Diseño de un material formado por un número de capas aislantes



Ejemplo

Movimiento = Permutación

2 6 7 3 4 5 1

Operador = Intercambio

2 5 7 3 4 6 1

Estructura de memoria a corto plazo

1	2	3	4	5	6	7
2						
3						
4						
5						
6						
7						

Ejemplo

Movimiento = Permutación

2 6 7 3 4 5 1

Operador = Intercambio

2 5 7 3 4 6 1

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

Estructura de memoria a corto plazo

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Ejemplo

ITERACIÓN 0

Solución actual

2 5 7 3 4 6 1

Estructura tabú

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Ejemplo

ITERACIÓN 0

Solución actual

2	5	7	3	4	6	1
---	---	---	---	---	---	---

Vaislam

10

Estructura tabú

2	3	4	5	6	7
1					
2					
3					
4					
5					
6					

Ejemplo

ITERACIÓN 0

Solución actual

2	5	7	3	4	6	1	Vaislam	10
---	---	---	---	---	---	---	---------	----

Operador	Valor
5, 4	6
7, 4	4
3, 6	2
2, 3	0

Estructura tabú

2	3	4	5	6	7
1					
	2				
		3			
			4		
				5	
					6

Ejemplo

ITERACIÓN 0

Solución actual

2 5 7 3 4 6 1	10
---------------	----

Vaislam

Operador	Valor	
5, 4	6	*
7, 4	4	
3, 6	2	
2, 3	0	

Estructura tabú

1	2	3	4	5	6	7
2						
3						
4						
5						
6						
7						

Ejemplo

ITERACIÓN 1

Solución actual

2 4 7 3 5 6 1

Estructura tabú

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Tenencia tabú = 3

Ejemplo

ITERACIÓN 1

SOLUCIONES TABÚ**2 5 7 3 4 6 1****Solución actual****2 4 7 3 5 6 1****Estructura tabú**

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Tenencia tabú = 3

Ejemplo

ITERACIÓN 1

SOLUCIONES TABÚ**2 5 7 3 4 6 1****Solución actual****2 4 7 3 5 6 1****Vaislam****16****Estructura tabú**

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Tenencia tabú = 3

Ejemplo

ITERACIÓN 1

SOLUCIONES TABÚ

2 5 7 3 4 6 1

Solución actual

2 4 7 3 5 6 1

Vaislam

16

Operador	Valor
3, 1	2
2, 3	1
7, 1	-2
6, 1	-4

Estructura tabú

1	2	3	4	5	6	7
2						
3						
4						
5						
6						
7						

Tenencia tabú = 3

Ejemplo

ITERACIÓN 1

SOLUCIONES TABÚ

2 5 7 3 4 6 1

Solución actual

Vaislam

2 4 7 3 5 6 1

16

Operador	Valor	
3, 1	2	*
2, 3	1	
7, 1	-2	
6, 1	-4	

Estructura tabú

1	2	3	4	5	6	7
1	2	3	4	5	6	7
2	3	4	5	6	7	
3	4	5	6	7		
4	5	6	7			
5	6	7				
6	7					

Tenencia tabú = 3

Ejemplo

ITERACIÓN 2

SOLUCIONES TABÚ

2 5 7 3 4 6 1

Solución actual

2 4 7 1 5 6 3

Vaislam

18

Estructura tabú

	2	3	4	5	6	7
1		3				
2						
3						
4					3-1=2	
5						
6						

Ejemplo

SOLUCIONES TABÚ

2 4 7 3 5 6 1

2 5 7 3 4 6 1

Solución actual

2 4 7 1 5 6 3

Vaislam

18

ITERACIÓN 2

Estructura tabú

	2	3	4	5	6	7
1		3				
2						
3						
4					3-1=2	
5						
6						

Ejemplo

ITERACIÓN 2

SOLUCIONES TABÚ

2	4	7	3	5	6	1
2	5	7	3	4	6	1

Solución actual

2	4	7	<u>1</u>	5	6	3
Vaislam						18

Operador	Valor
1, 3	-2
2, 4	-4
7, 6	-6
4, 5	-7

Tabú

*

Tabú

Estructura tabú

1	2	3	4	5	6	7
2						
3						
4						
5						

3-1=2

Ejemplo

ITERACIÓN 3

SOLUCIONES TABÚ

2 4 7 3 5 6 1

Solución actual

4 2 7 1 5 6 3

Vaislam

14

Estructura tabú

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						
7						

Ejemplo

SOLUCIONES TABÚ

2	4	7	1	5	6	3
2	4	7	3	5	6	1

ITERACIÓN 3

Solución actual

4 2 7 1 5 6 3

Vaislam

14

Estructura tabú

	2	3	4	5	6	7
1		2				
2			3			
3				4	1	
4					5	
5						6

Ejemplo

ITERACIÓN 3

SOLUCIONES TABÚ

2	4	7	1	5	6	3
2	4	7	3	5	6	1

Solución actual

4	2	7	1	5	6	3
						14

Vaislam

Operador	Valor
4, 5	6
5, 3	2
7, 1	0
1, 3	-3

Tabú*

Tabú

Estructura tabú

1	2	3	4	5	6	7
2	1	3				
3	2	1	4	5	6	
4	3		1	2	3	
5		4	3	2	1	7
6			5	4	3	2
7				6	5	4

Ejemplo

ITERACIÓN 3

SOLUCIONES TABÚ

2 4 7 1 5 6 3

Solución actual

5 2 7 1 4 6 3

Vaislam

20

Estructura tabú

	2	3	4	5	6	7
1			1			
2			2			
3				3		
4					3	
5						5
6						

Ejemplo

SOLUCIONES TABÚ

4 2 7 1 5 6 3

2 4 7 1 5 6 3

Solución actual

5 2 7 1 4 6 3

Vaislam

20

ITERACIÓN 3

Estructura tabú

	2	3	4	5	6	7
1		1				
2			2			
3				3		
4					3	
5						5
6						

Ejemplo

- Diagonal superior representa la lista tabú, y la inferior la frecuencia de aparición de las capas 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 a corto plazo

	1	2	3	4	5	6	7
1							
2							
3							
4							
5							
6							
7							

Memoria a largo plazo

The grid illustrates the short-term and long-term memory structures. The top row (1) and bottom row (7) are labeled with indices 1 through 7. The grid itself has a 7x7 structure. The values 1, 2, and 3 are distributed across the grid. A diagonal of 1s runs from index (1,1) to (7,7). A diagonal of 2s runs from index (1,7) to (7,1). A diagonal of 3s runs from index (4,5) to (5,4). The background of the grid is red.

Problema del viajante de comercio

Formulación del problema

- **Generación de la solución inicial greedy** (Unidad 1.1)
- **Esquema de representación** Representación de orden mediante permutación {1,...,n}
- **Operador de generación de vecinos** seleccionar dos ciudades e intercambiarlas (2-opt)
- **Función objetivo** (minimización)

Formulación del problema

- **Listá tabú** almacenando las ciudades intercambiadas en el operador 2-opt
- **Estrategia de selección** de 3 movimientos aleatorios
- **Criterio de aspiración** se cumple si tenemos una solución tabú con mejor coste
- **En la memoria a largo plazo** almacenamos la frecuencia de aparición entre dos ciudades donde $i \rightarrow j = j \rightarrow i$

Resolución del problema

Excepciones:

- Se debe iniciar en la ciudad 0
- No se pueden repetir ciudades

	1	2	3	4	5	6	7	8	9	10
0	76	7	73	69	79	54	58	38	23	39
1	-	82	108	16	52	99	51	42	72	59
2	82	-	69	74	81	50	60	42	22	40
3	108	69	-	92	68	19	57	73	52	52
4	16	74	92	-	36	84	35	31	60	45
5	52	81	68	36	-	68	21	46	60	41
6	99	50	19	84	68	-	52	60	35	40
7	51	60	57	35	21	52	-	28	39	20
8	42	42	73	31	46	60	28	-	30	21
9	72	22	52	60	60	35	39	30	-	19
10	59	40	52	45	41	40	20	21	19	-

Resolución del problema

Solución Greedy

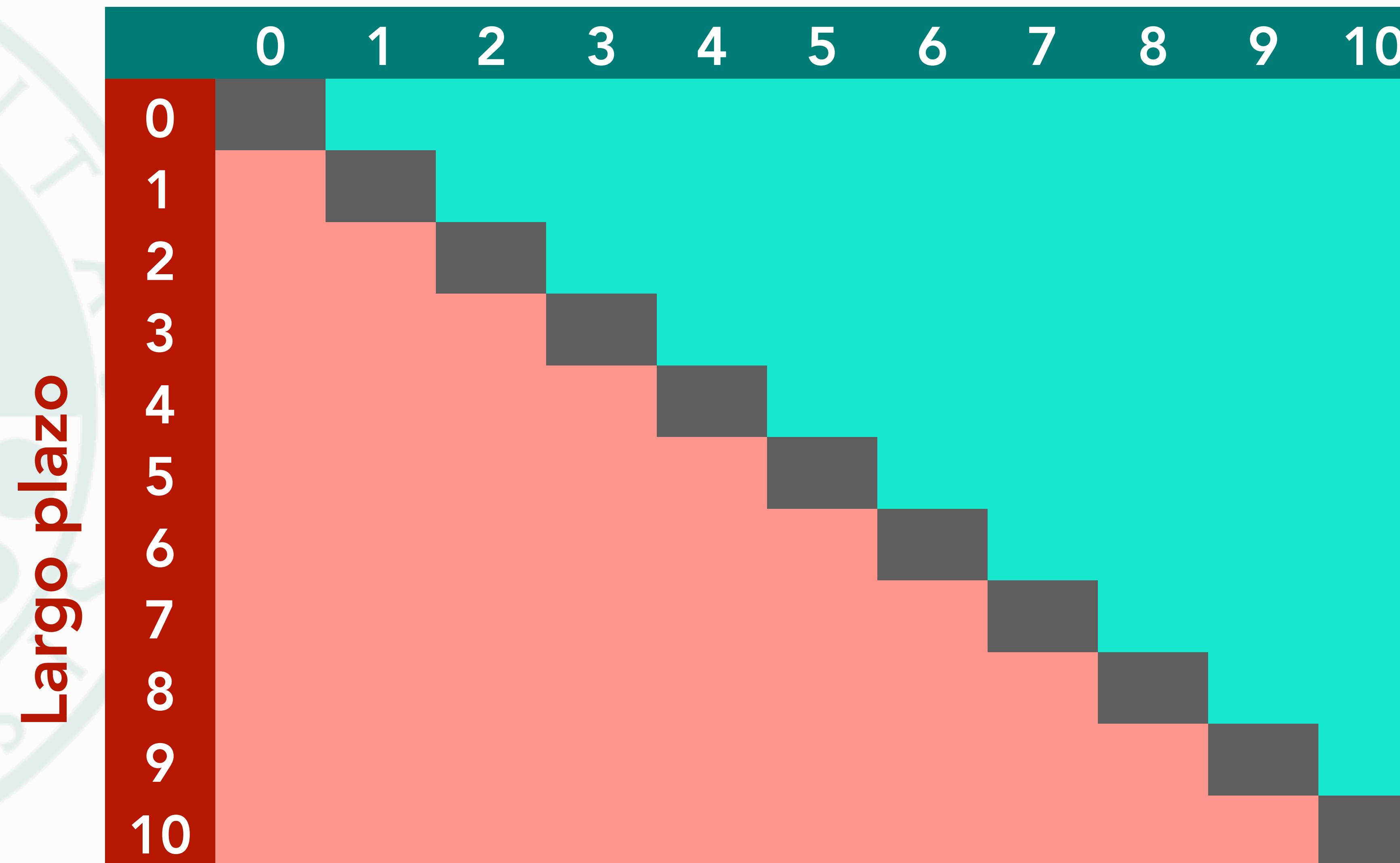
0 2 9 10 7 5 4 1 8 6 3

Coste: 370

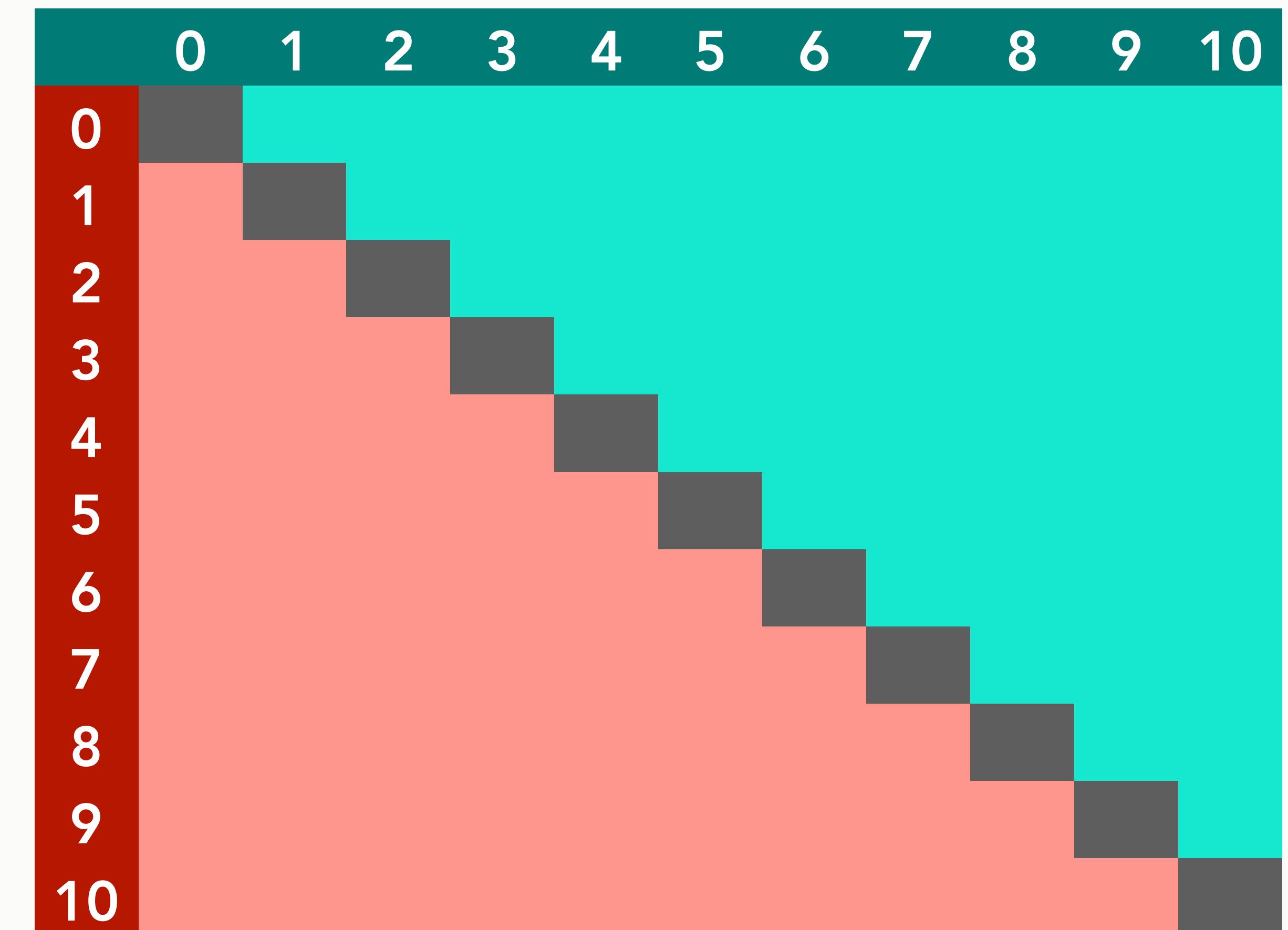
	1	2	3	4	5	6	7	8	9	10
0	76	7	73	69	79	54	58	38	23	39
1	-	82	108	16	52	99	51	42	72	59
2	82	-	69	74	81	50	60	42	22	40
3	108	69	-	92	68	19	57	73	52	52
4	16	74	92	-	36	84	35	31	60	45
5	52	81	68	36	-	68	21	46	60	41
6	99	50	19	84	68	-	52	60	35	40
7	51	60	57	35	21	52	-	28	39	20
8	42	42	73	31	46	60	28	-	30	21
9	72	22	52	60	60	35	39	30	-	19
10	59	40	52	45	41	40	20	21	19	-

Resolución del problema

Corto plazo



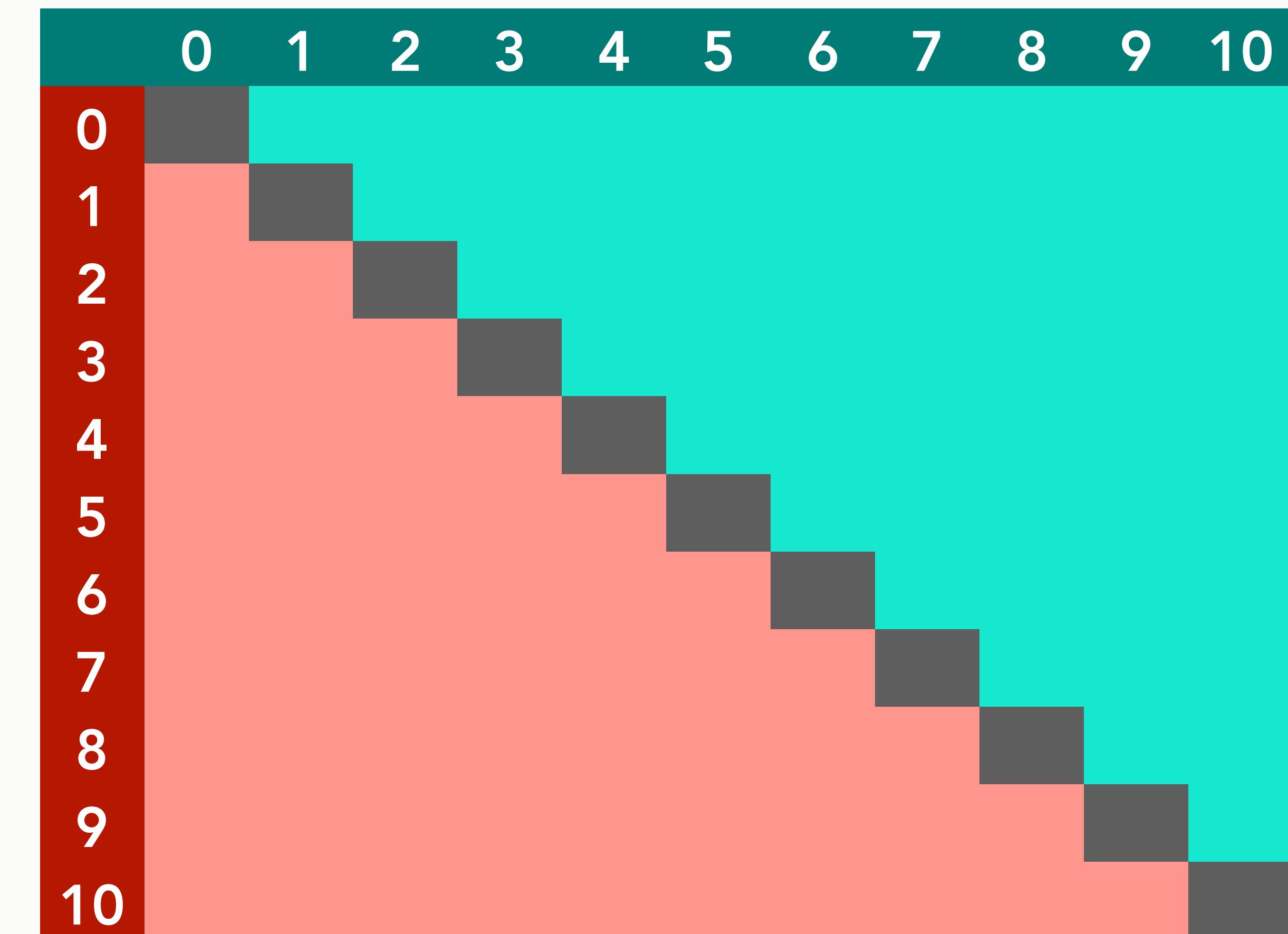
Resolución del problema

Iteración 1

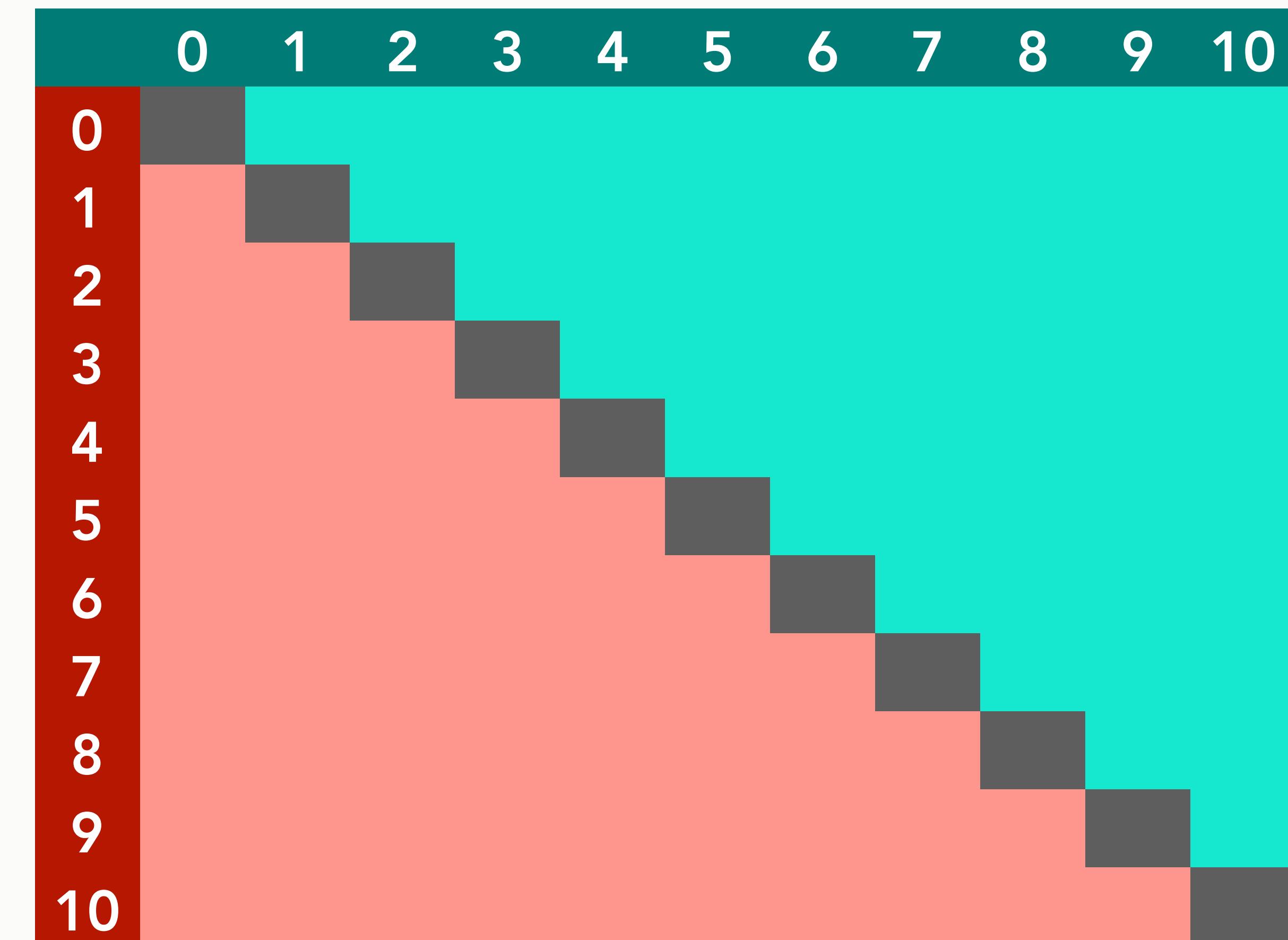
Resolución del problema

Iteración 1

S: (0 2 9 10 7 5 4 1 8 6 3) → 370

LT = {Ø}

Resolución del problema

Iteración 1 $S: (0 \ 2 \ 9 \ 10 \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 370$ $E^*=\{(9 \ 10), (1 \ 7), (5 \ 8)\}$ $LT=\{\underline{\emptyset}\}$ 

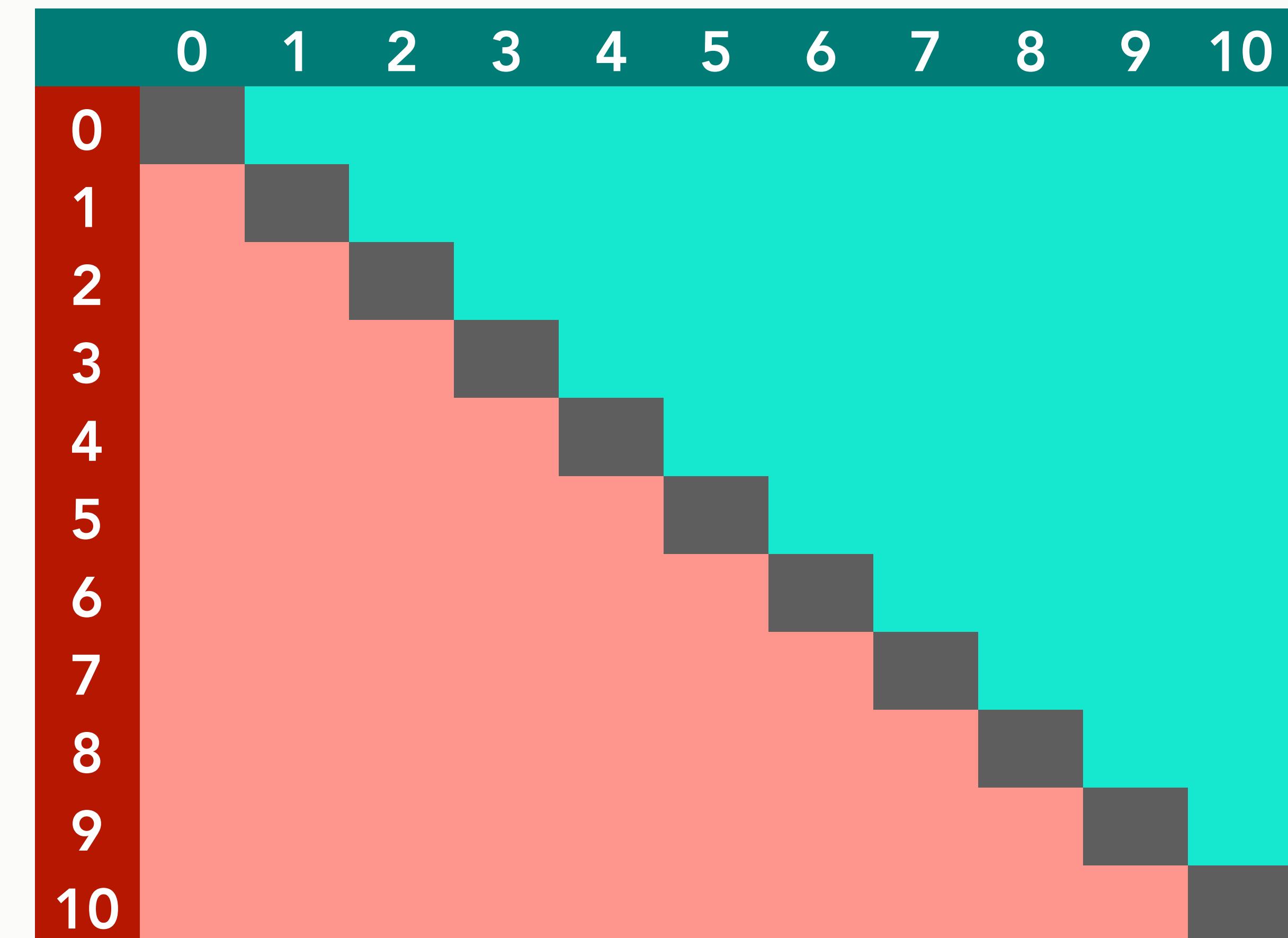
Resolución del problema

Iteración 1 $S: (0 \ 2 \ 9 \ 10 \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 370$

$E^* = \{(9 \ 10), (1 \ 7), (5 \ 8)\}$

 $S^*: (0 \ 2 \ \underline{10 \ 9} \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 407$

$LT = \{\emptyset\}$



Resolución del problema

Iteración 1

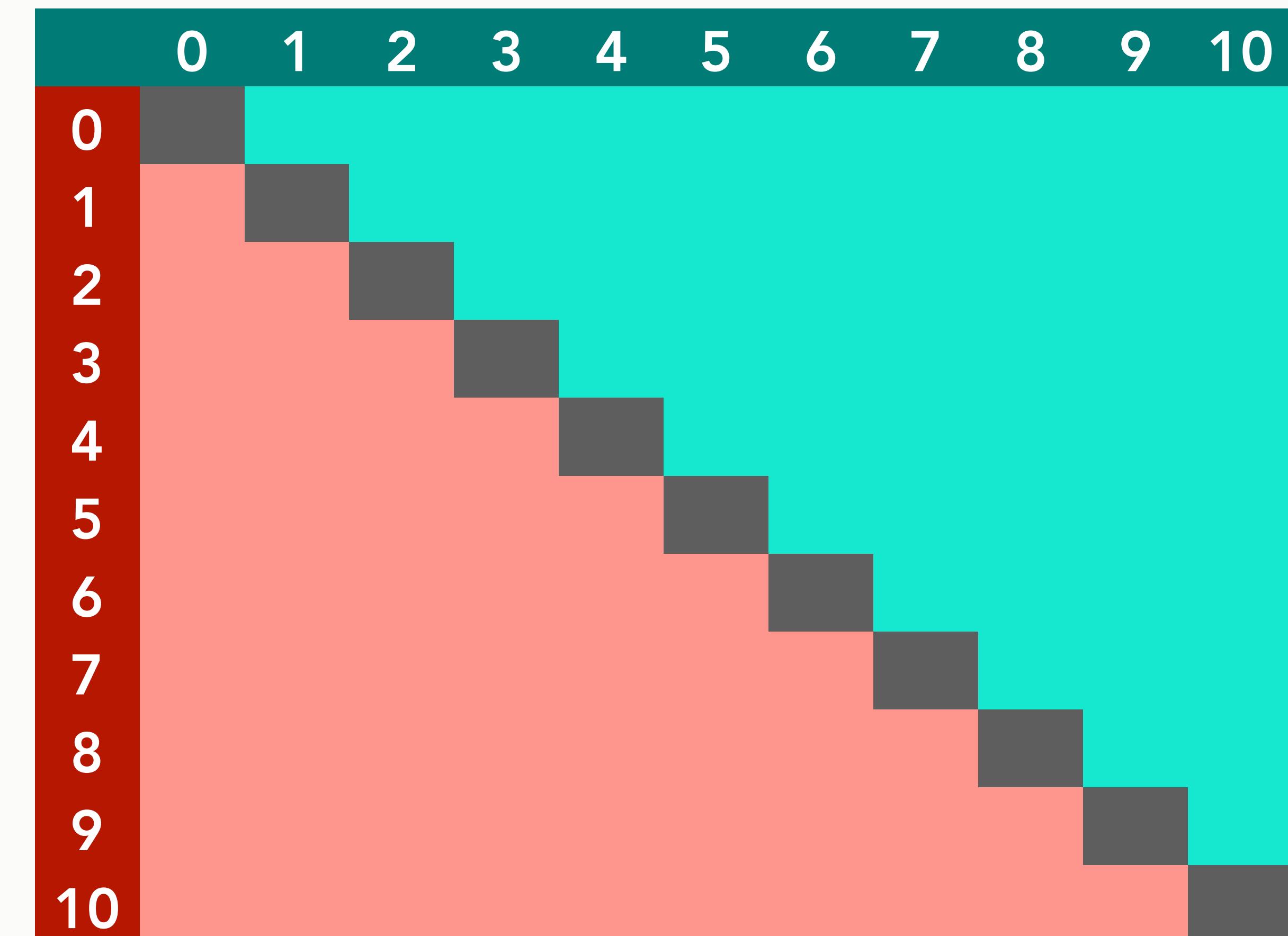
$S: (0 \ 2 \ 9 \ 10 \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 370$

$E^* = \{(9 \ 10), (1 \ 7), (5 \ 8)\}$

$S^*: (0 \ 2 \ \underline{10 \ 9} \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 407$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 5 \ 4 \ \underline{7} \ 8 \ 6 \ 3) \rightarrow 445$

$LT = \{\emptyset\}$



Resolución del problema

Iteración 1

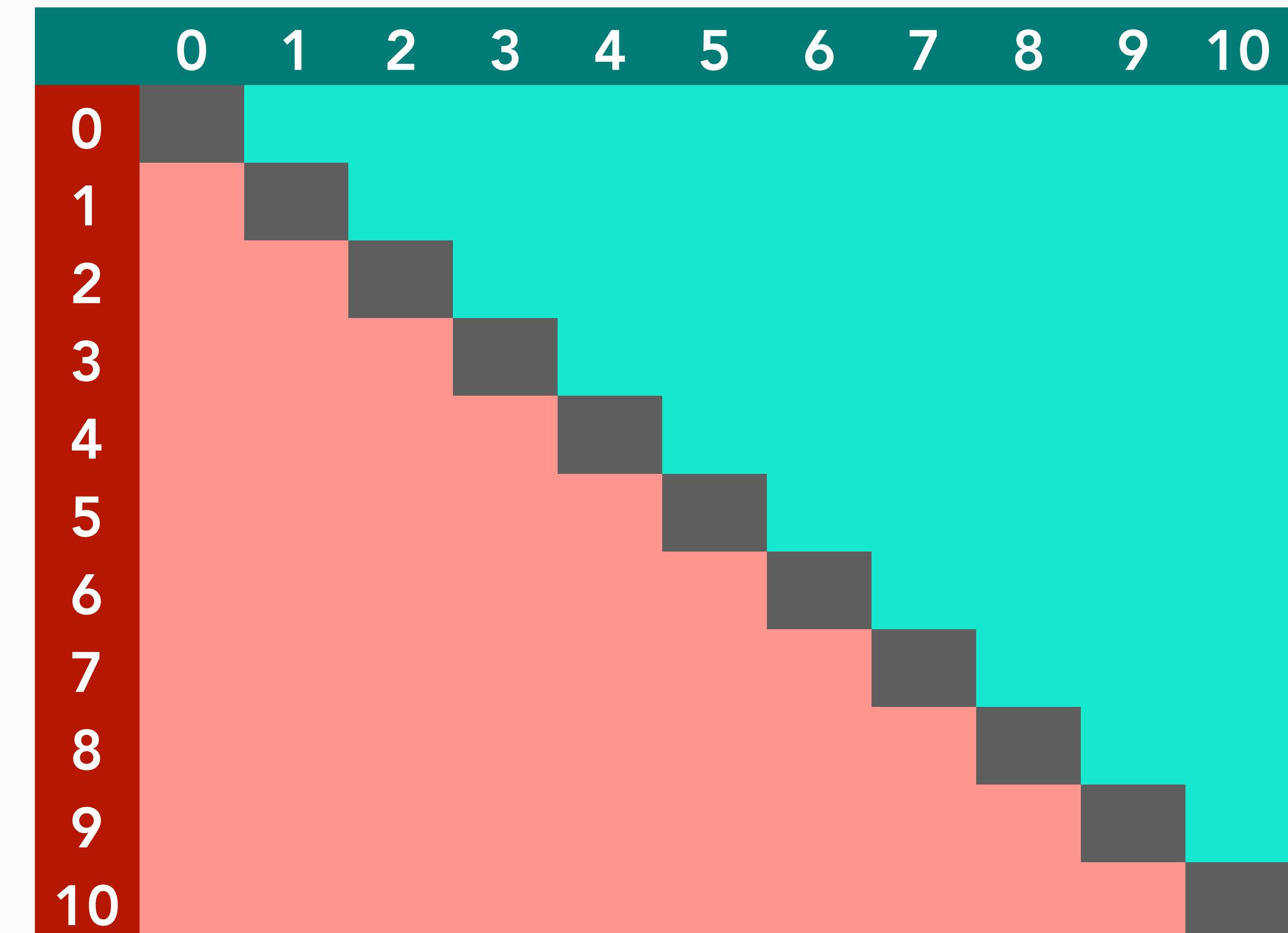
$S: (0 \ 2 \ 9 \ 10 \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 370$

$$E^* = \{(9 \ 10), (1 \ 7), (5 \ 8)\}$$

$S^*: (0 \ 2 \ \underline{10 \ 9} \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 407$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1 \ 5} \ 4 \ \underline{7 \ 8} \ 6 \ 3) \rightarrow 445$

$LT = \{\emptyset\}$



Resolución del problema

Iteración 1

$S: (0 \ 2 \ 9 \ 10 \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 370$

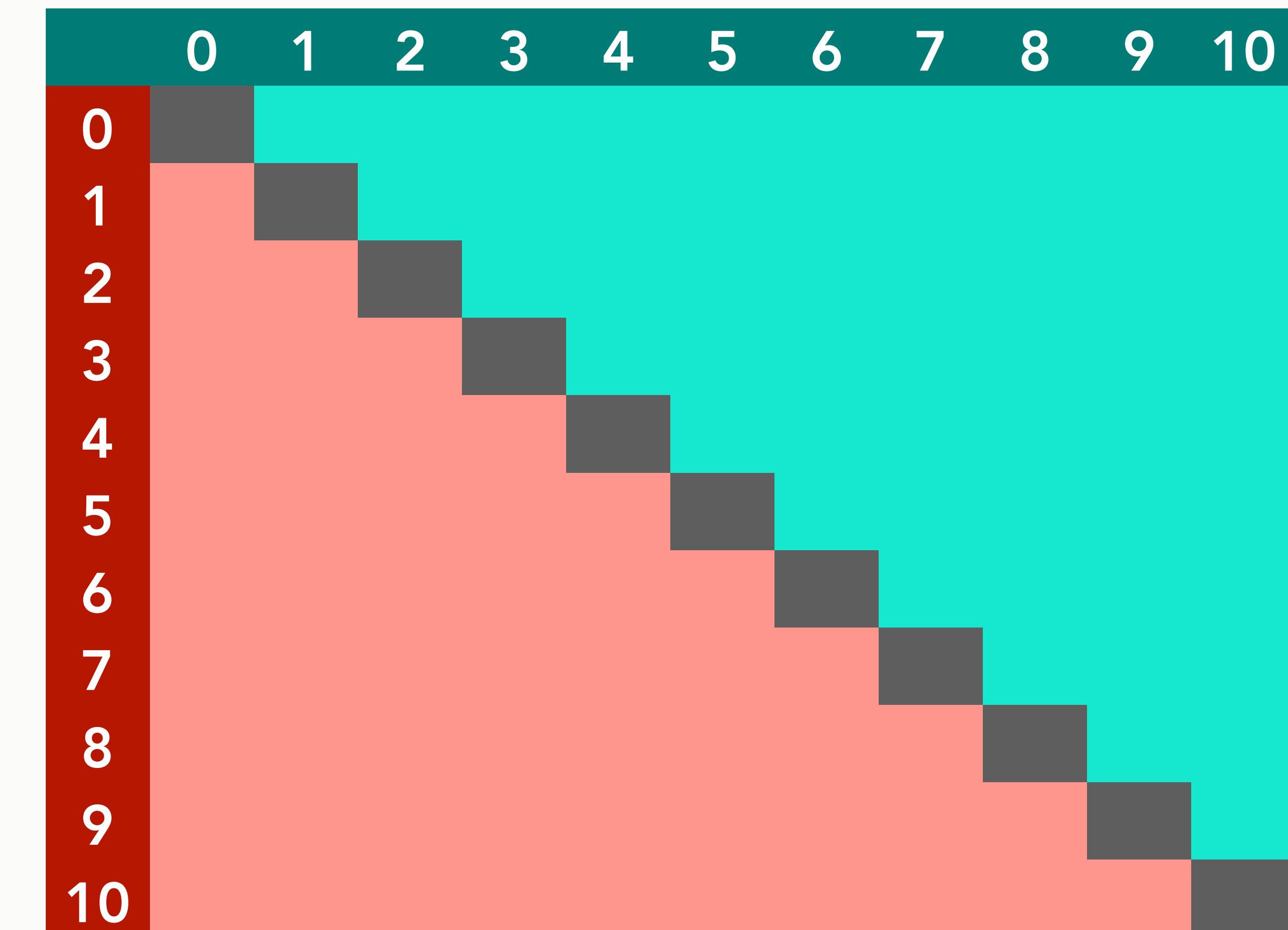
$$E^* = \{(9 \ 10), (1 \ 7), (5 \ 8)\}$$

$S^*: (0 \ 2 \ \underline{10 \ 9} \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 407$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1 \ 5} \ 4 \ \underline{7 \ 8} \ 6 \ 3) \rightarrow 445$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ \underline{8} \ 4 \ 1 \ \underline{5} \ 6 \ 3) \rightarrow 390$

$$LT = \{\emptyset\}$$



Resolución del problema

$S: (0 \ 2 \ 9 \ 10 \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 370$

$$E^* = \{(9 \ 10), (1 \ 7), (5 \ 8)\}$$

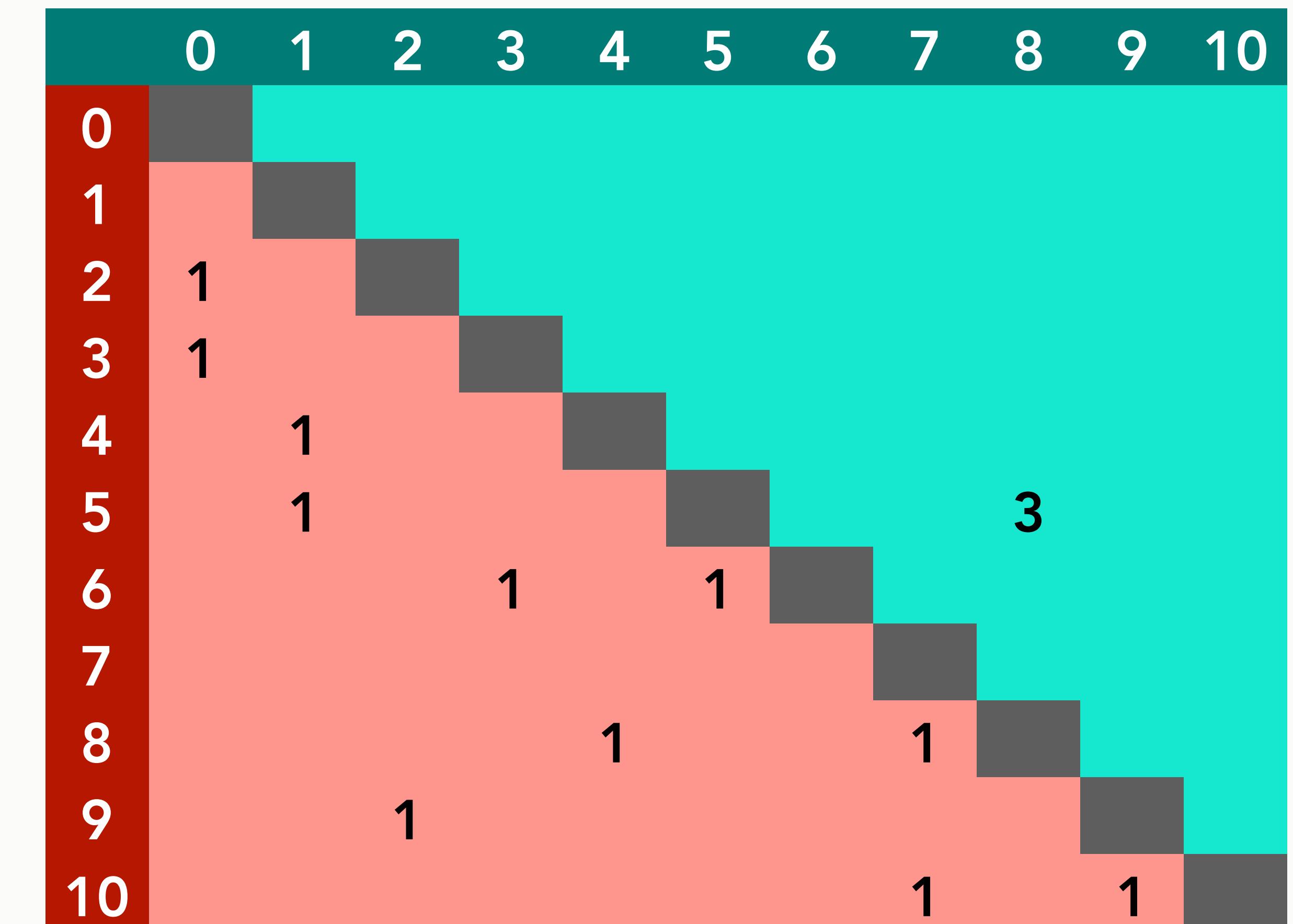
$S^*: (0 \ 2 \ \underline{10 \ 9} \ 7 \ 5 \ 4 \ 1 \ 8 \ 6 \ 3) \rightarrow 407$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 5 \ 4 \ \underline{7} \ 8 \ 6 \ 3) \rightarrow 445$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ \underline{8} \ 4 \ 1 \ \underline{5} \ 6 \ 3) \rightarrow 390$

$$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3)\}$$

Iteración 1



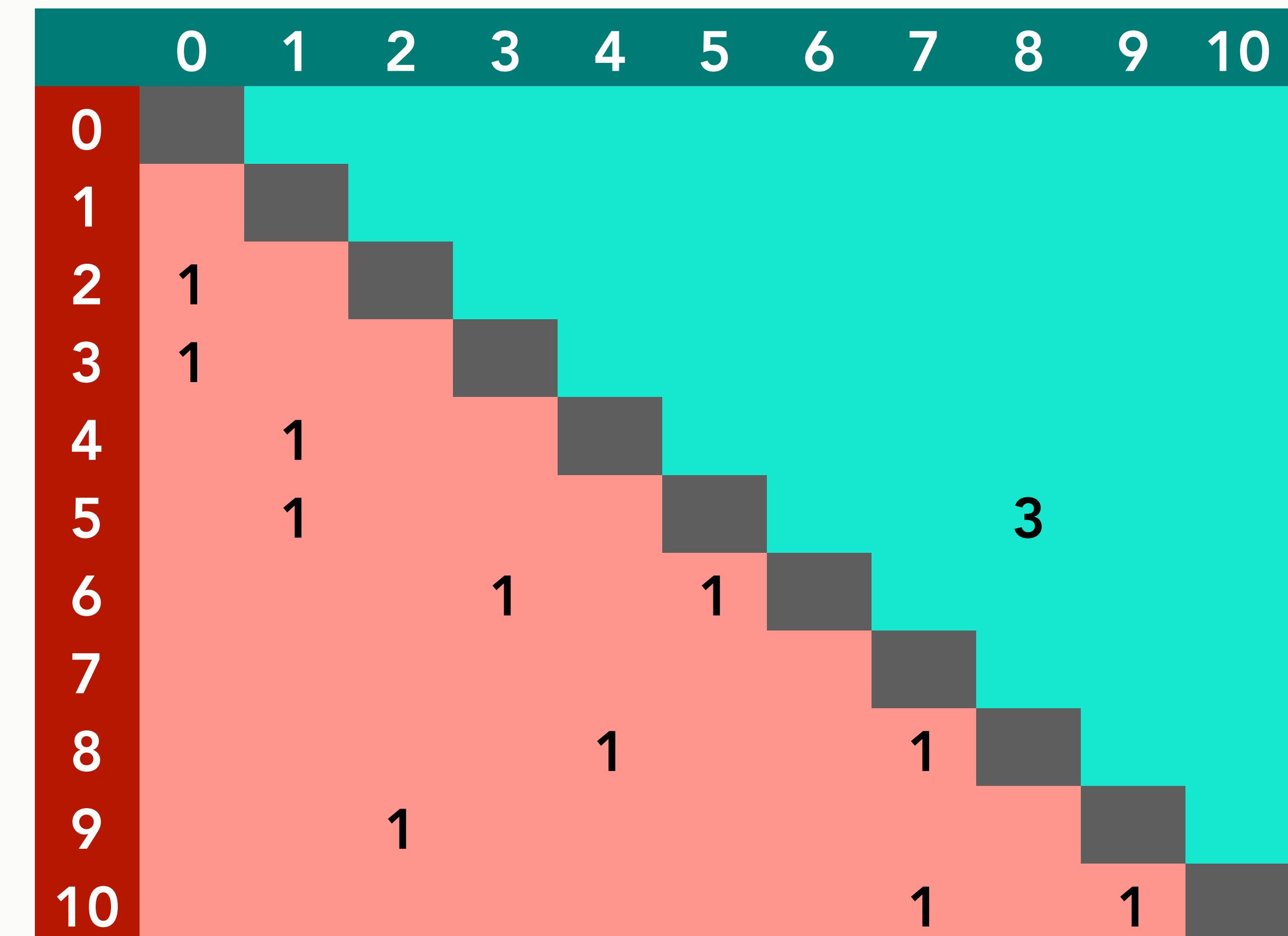
Resolución del problema

Iteración 2

$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3) \rightarrow 390$

$E^*=\{(7 \ 1), (4 \ 5), (3 \ 6)\}$

$LT=\{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3)\}$



Resolución del problema

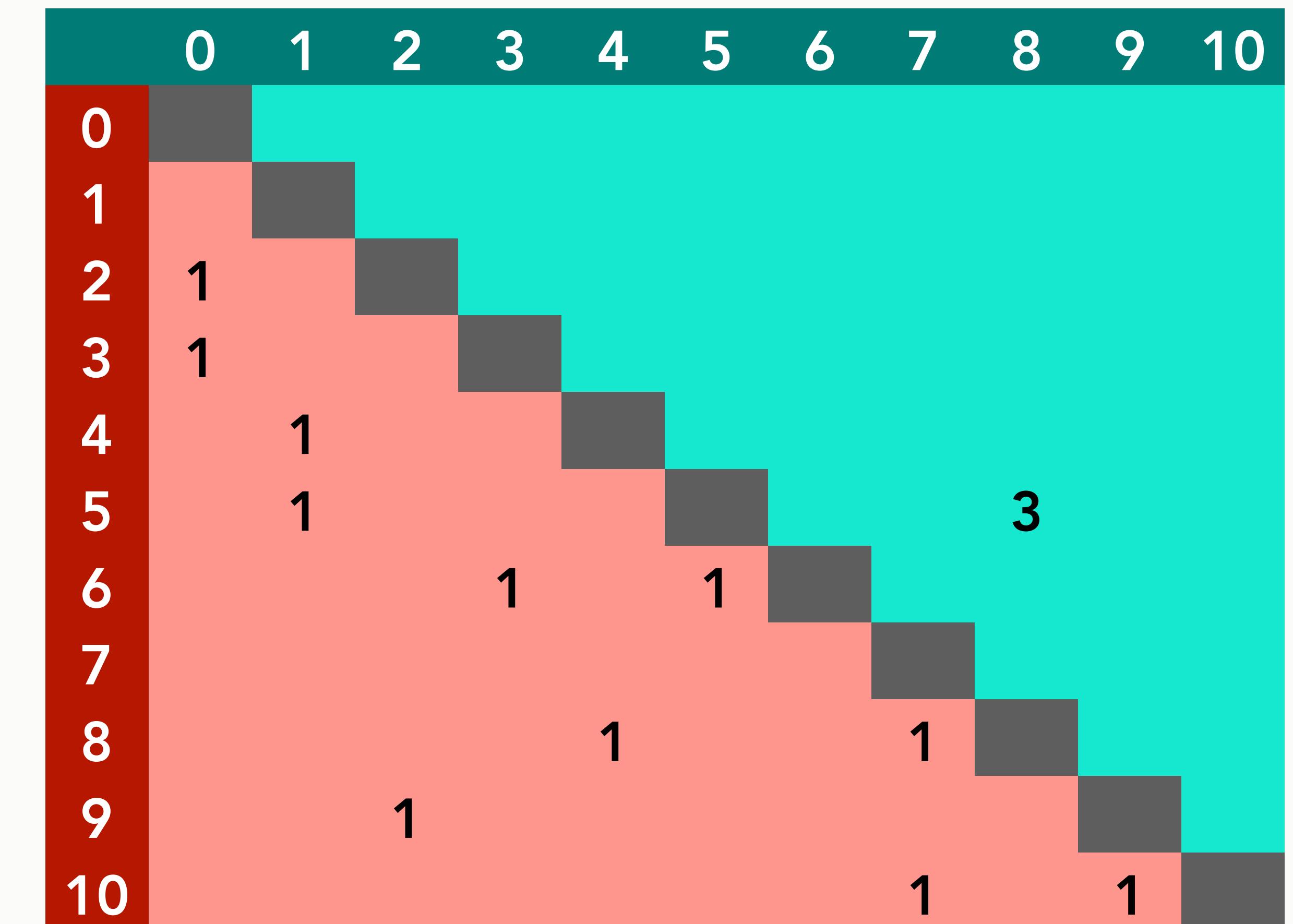
$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3) \rightarrow 390$

$$E^* = \{(7 \ 1), (4 \ 5), (3 \ 6)\}$$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 8 \ 4 \ \underline{7} \ 5 \ 6 \ 3) \rightarrow 453$

$$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3)\}$$

Iteración 2



Resolución del problema

Iteración 2

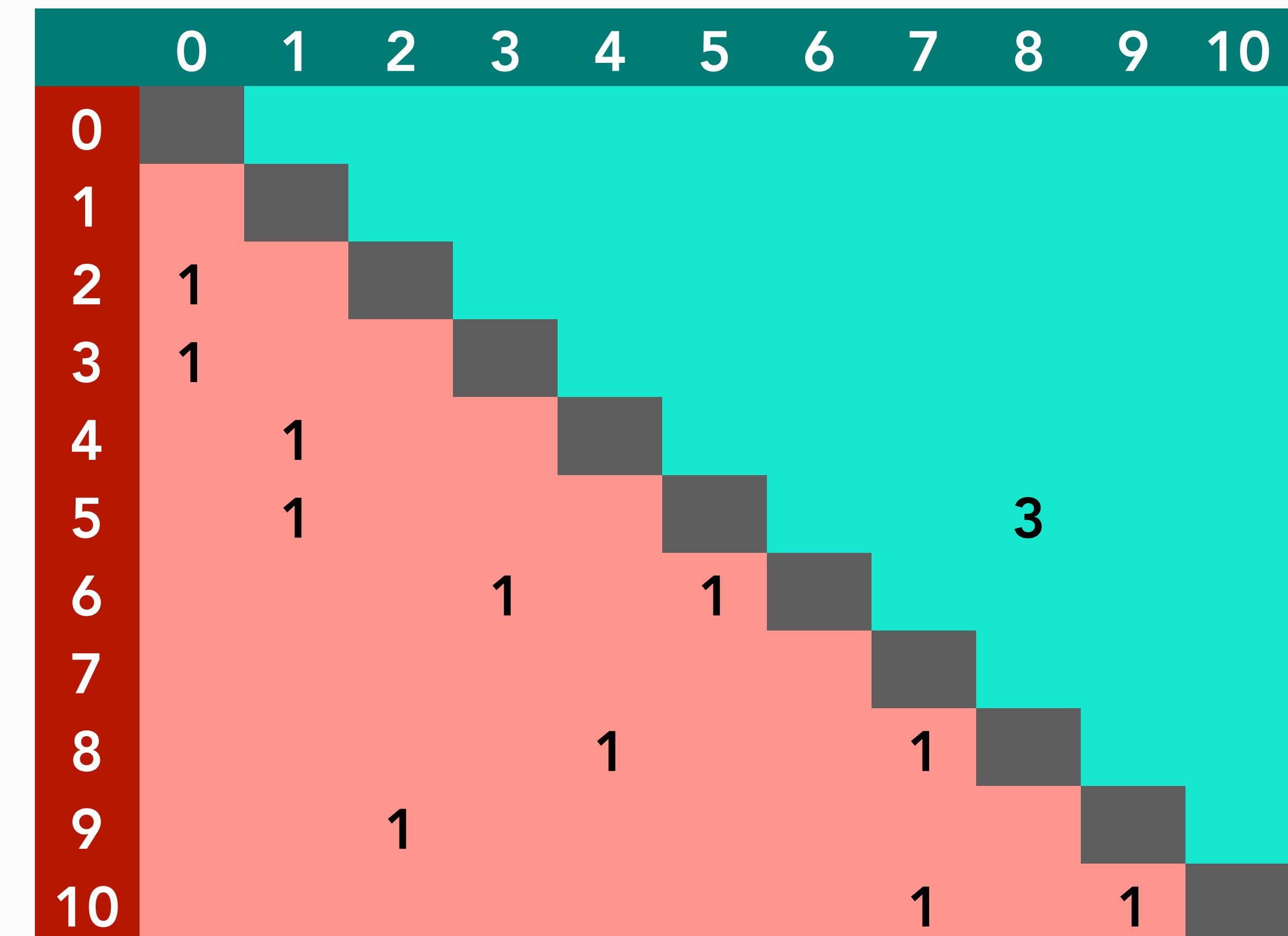
$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3) \rightarrow 390$

$E^* = \{(7 \ 1), (4 \ 5), (3 \ 6)\}$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 8 \ 4 \ \underline{7} \ 5 \ 6 \ 3) \rightarrow 453$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ \underline{5} \ 1 \ \underline{4} \ 6 \ 3) \rightarrow 421$

$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3)\}$



Resolución del problema

Iteración 2

$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3) \rightarrow 390$

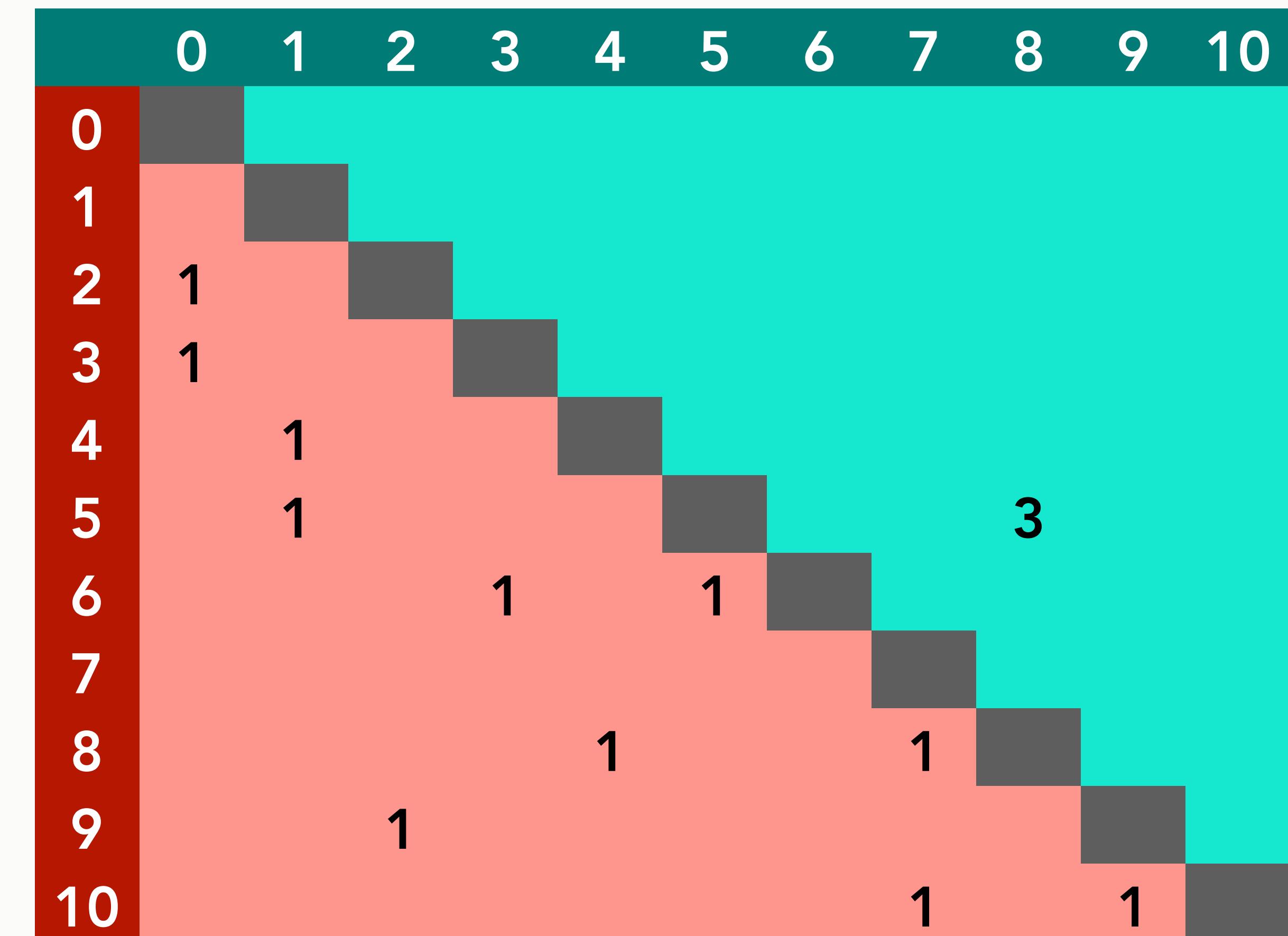
$E^* = \{(7 \ 1), (4 \ 5), (3 \ 6)\}$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 8 \ 4 \ \underline{7} \ 5 \ 6 \ 3) \rightarrow 453$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ \underline{5} \ 1 \ \underline{4} \ 6 \ 3) \rightarrow 421$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ \underline{3} \ \underline{6}) \rightarrow 381$

$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3)\}$



Resolución del problema

Iteración 2

$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3) \rightarrow 390$

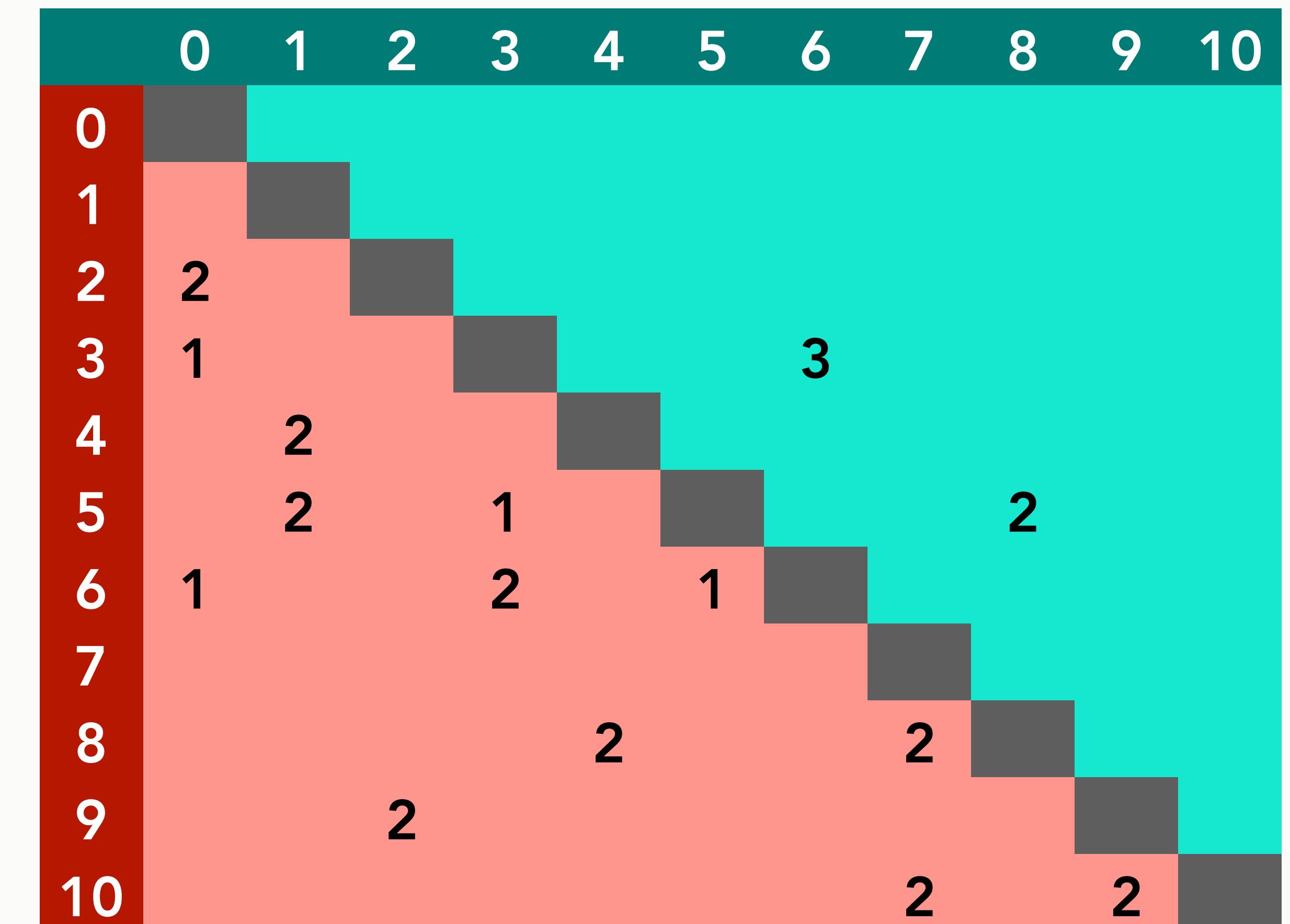
$E^* = \{(7 \ 1), (4 \ 5), (3 \ 6)\}$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 8 \ 4 \ \underline{7} \ 5 \ 6 \ 3) \rightarrow 453$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ \underline{5} \ 1 \ \underline{4} \ 6 \ 3) \rightarrow 421$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ \underline{3} \ \underline{6}) \rightarrow 381$

$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3), (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6)\}$

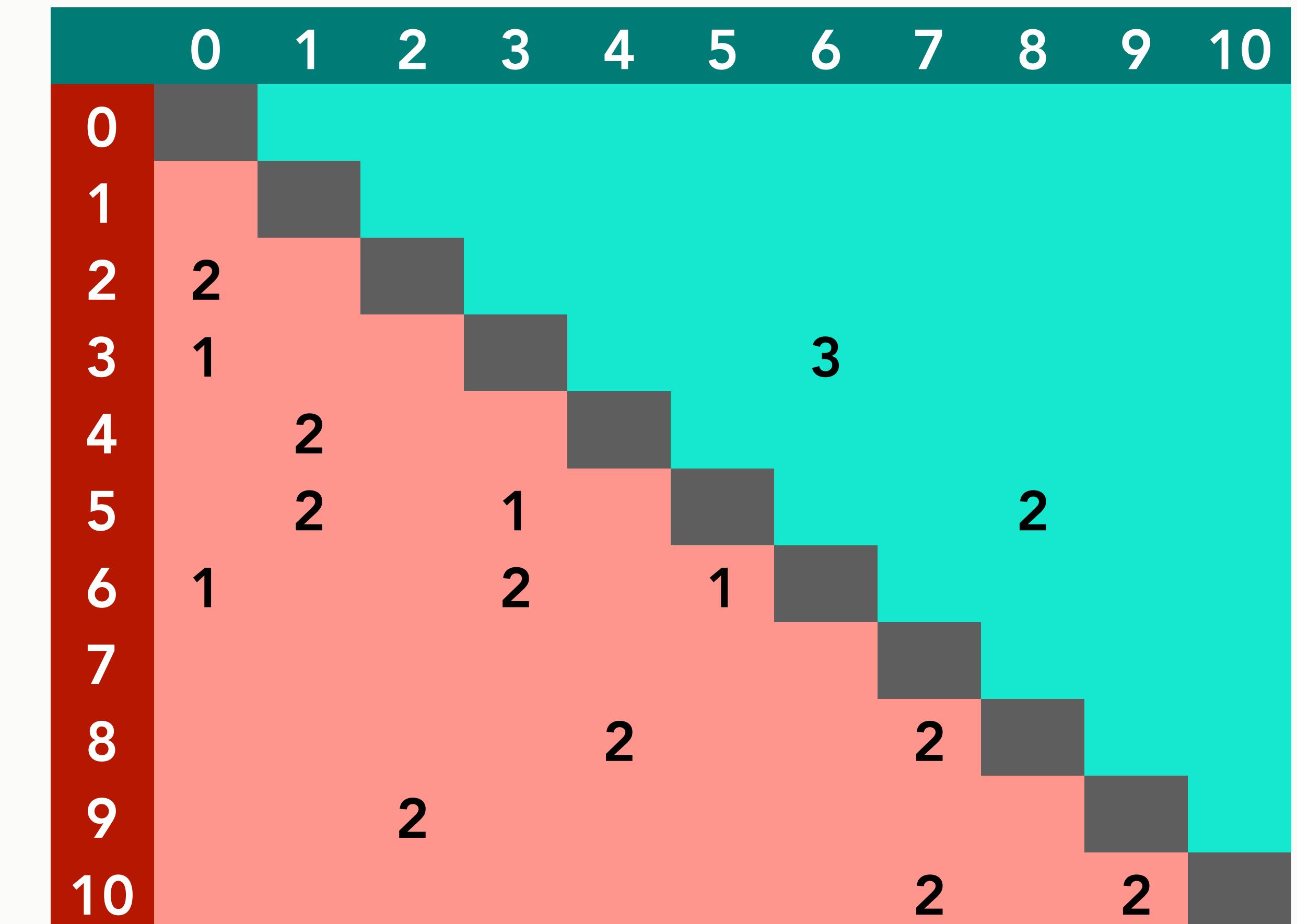


Resolución del problema

Iteración 3

$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6) \rightarrow 381$

$E^*=\{(5 \ 7), (7 \ 1), (6 \ 3)\}$



$LT=\{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3), (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6)\}$

Resolución del problema

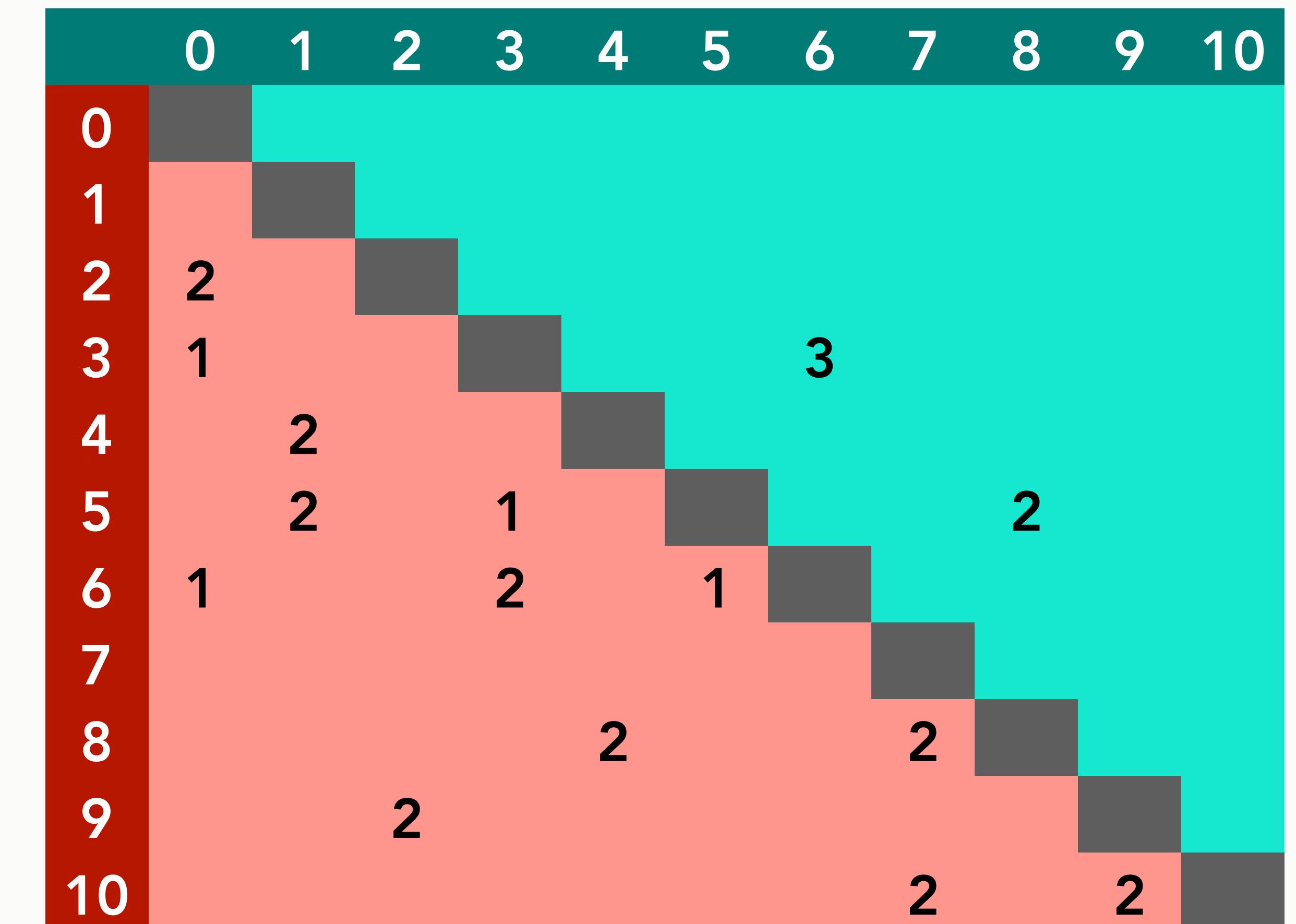
$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6) \rightarrow 381$

$E^* = \{(5 \ 7), (7 \ 1), (6 \ 3)\}$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{5} \ 8 \ 4 \ 1 \ \underline{7} \ 3 \ 6) \rightarrow 408$

$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3), (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6)\}$

Iteración 3



Resolución del problema

Iteración 3

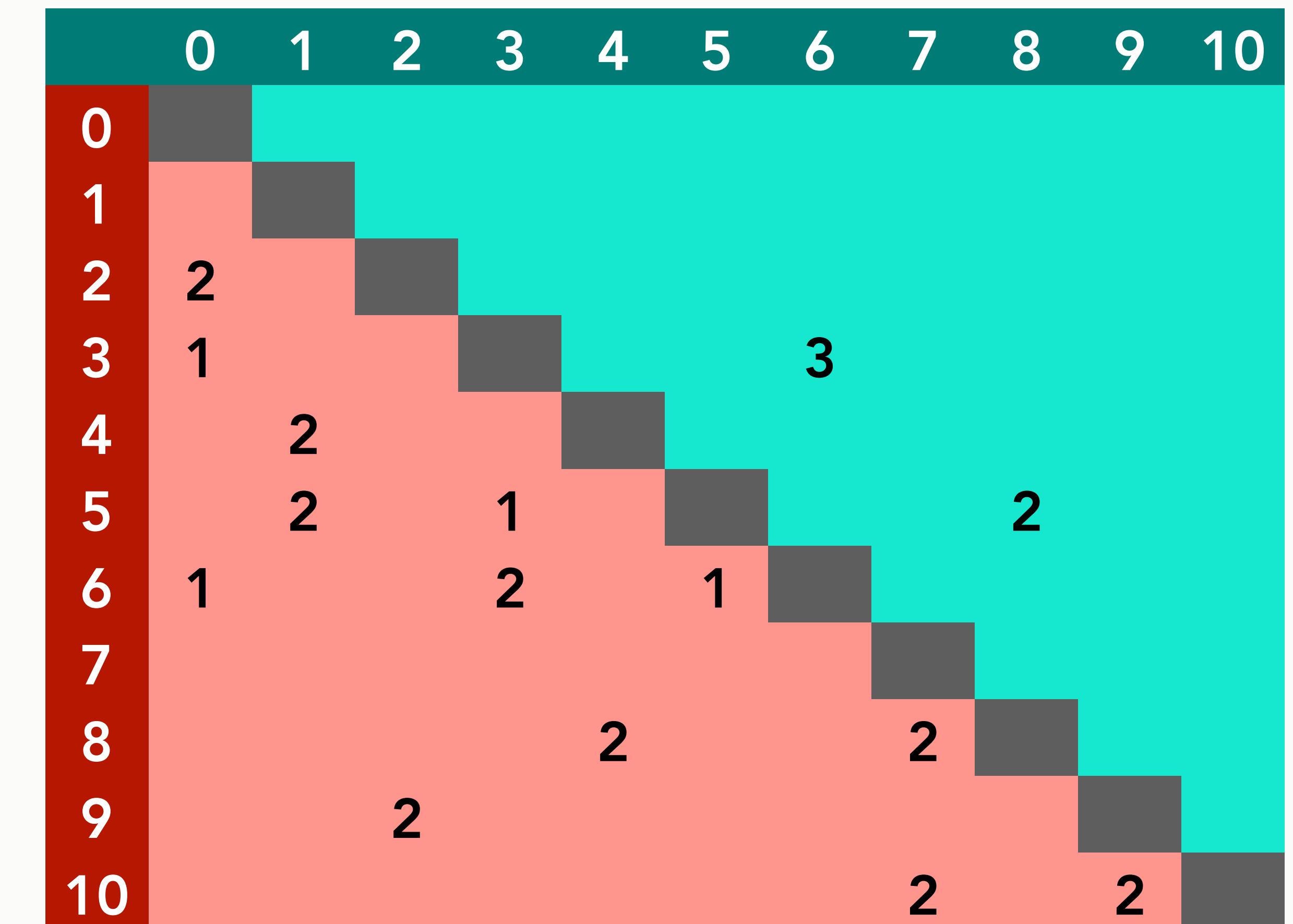
$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6) \rightarrow 381$

$E^* = \{(5 \ 7), (7 \ 1), (6 \ 3)\}$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{5} \ 8 \ 4 \ 1 \ \underline{7} \ 3 \ 6) \rightarrow 408$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 8 \ 4 \ \underline{7} \ 5 \ 3 \ 6) \rightarrow 422$

$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3), (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6)\}$



Resolución del problema

Iteración 3

$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6) \rightarrow 381$

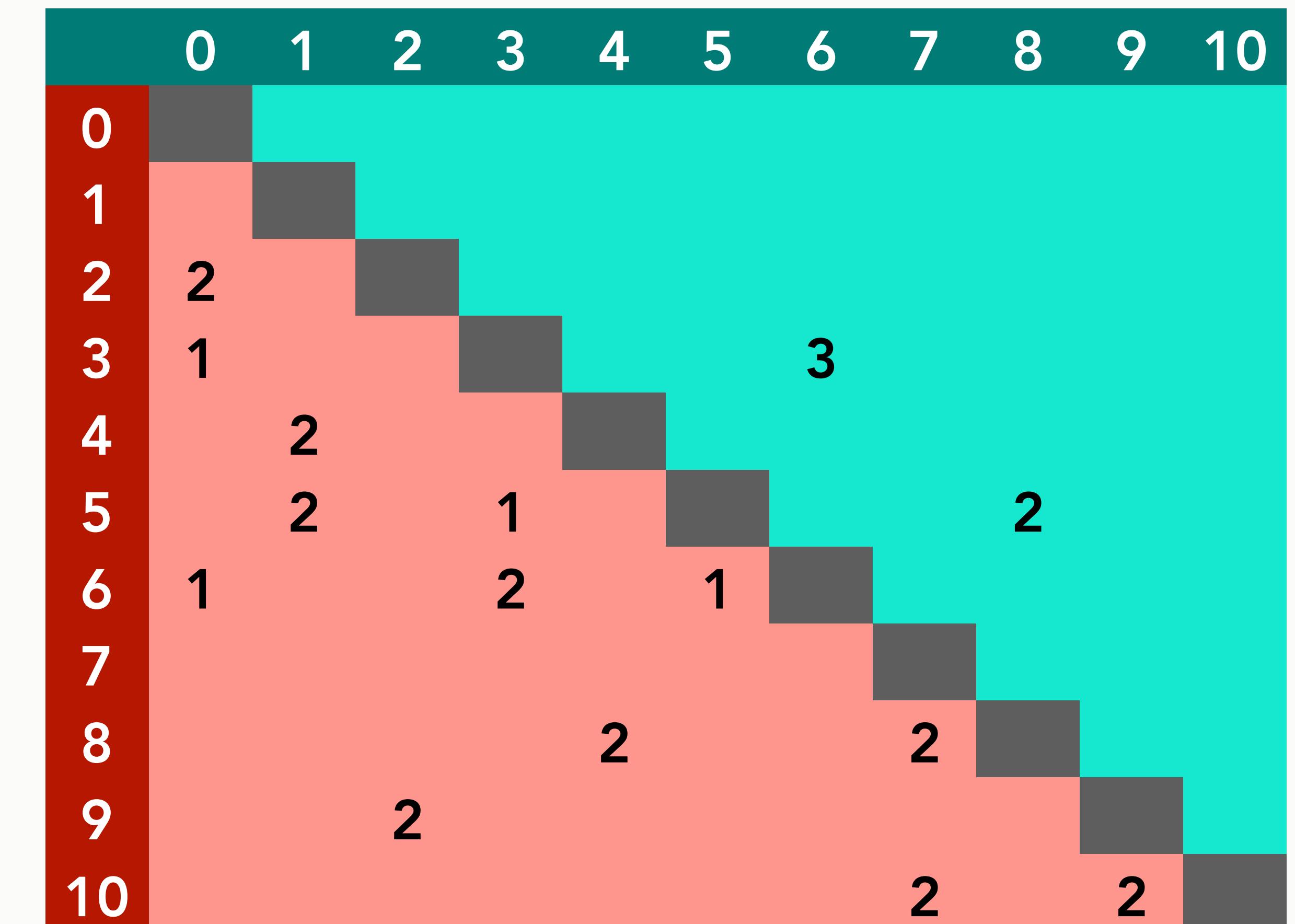
$E^* = \{(5 \ 7), (7 \ 1), (6 \ 3)\}$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{5} \ 8 \ 4 \ 1 \ \underline{7} \ 3 \ 6) \rightarrow 408$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 8 \ 4 \ \underline{7} \ 5 \ 3 \ 6) \rightarrow 422$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ \underline{6} \ \underline{3})$

$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3), (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6)\}$



Resolución del problema

Iteración 3

$S: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6) \rightarrow 381$

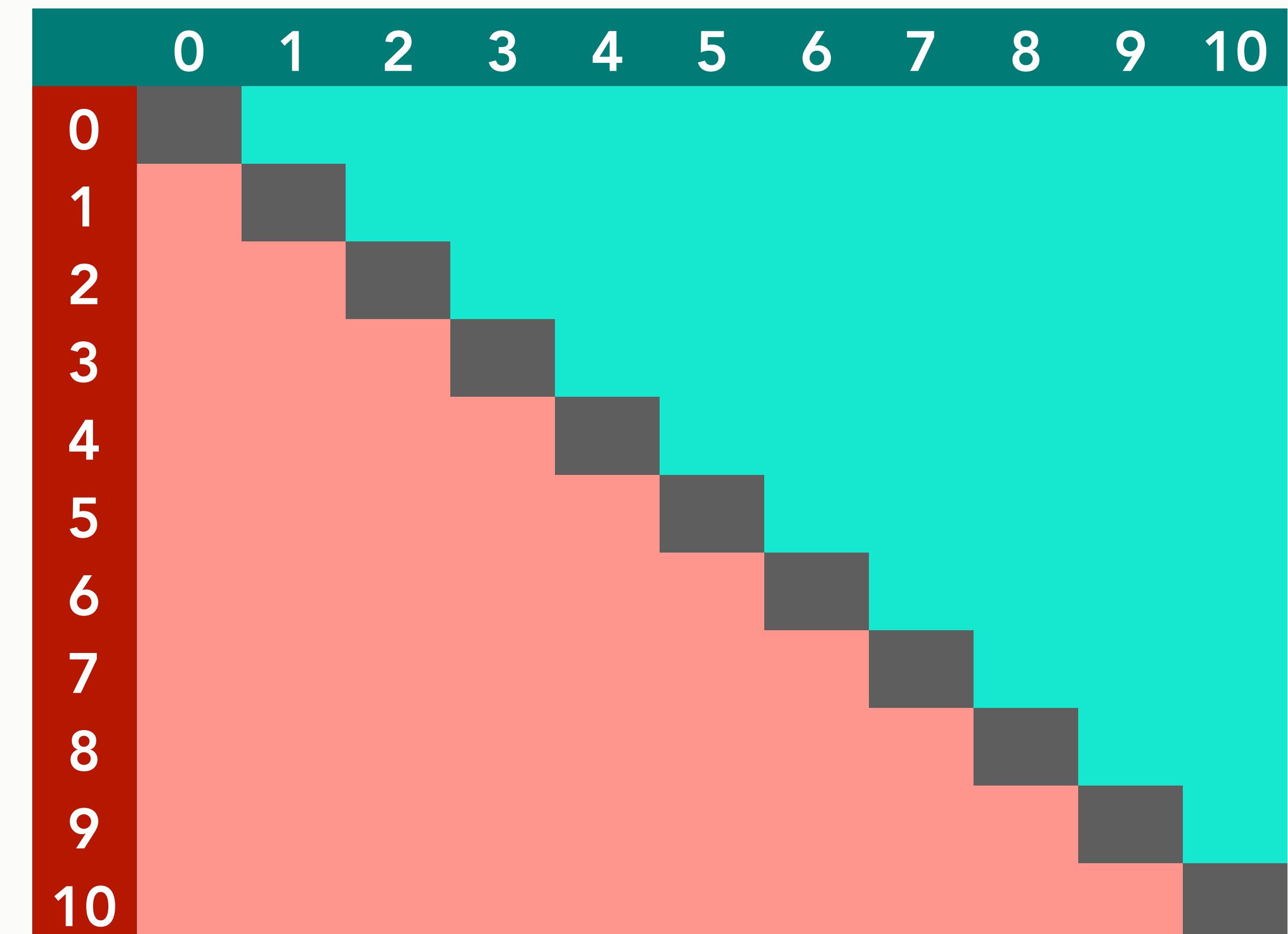
$E^* = \{(5 \ 7), (7 \ 1), (6 \ 3)\}$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{5} \ 8 \ 4 \ 1 \ \underline{7} \ 3 \ 6) \rightarrow 408$

$S^*: (0 \ 2 \ 9 \ 10 \ \underline{1} \ 8 \ 4 \ \underline{7} \ 5 \ 3 \ 6) \rightarrow 422$

$S^*: (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ \underline{6} \ \underline{3})$

$LT = \{(0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 6 \ 3), (0 \ 2 \ 9 \ 10 \ 7 \ 8 \ 4 \ 1 \ 5 \ 3 \ 6), (0 \ 2 \ 9 \ 10 \ 5 \ 8 \ 4 \ 1 \ 7 \ 3 \ 6)\}$



Resolución del problema

Iteración 4

S: (0 2 9 10 5 8 4 1 7 3 6) → 408

E*={. . .}

LT={(0 2 9 10 7 8 4 1 5 6 3), (0 2 9 10 7 8 4 1 5 3 6), (0 2 9 10 5 8 4 1 7 3 6)}



Metaheurísticas
Grado en Ingeniería Informática
Universidad de Jaén
Cristóbal J. Carmona
Curso 2023/2024

Esta obra está protegida con licencia
Creative Commons Atribución-NoComercial 4.0 Internacional

