

# Metaheurísticas

Unidad 1  
Introducción a las Metaheurísticas

**Tema 2: Métodos de búsqueda**

# Objetivos

- Analizar las características de las técnicas basadas en búsqueda local y las estructuras del entorno
- Entender el funcionamiento y estructura general de los algoritmos de optimización simples basados en búsquedas locales
- Ante un problema abordable mediante este tipo de técnica determinar la representación y operadores de vecindario más acordes
- Entender el concepto de algoritmo basado en poblaciones

# Bibliografía

- [Tal09] E.-G. Talbi. **Metaheuristics. From design to implementation.** Wiley, 2009
- [Bro12] J. Brownlee. **Clever Algorithms (Nature-Inspired Programming Recipes)**. 2012, Brownlee.

# Índice

1. Introducción. Búsquedas basadas en trayectorias y búsquedas basadas en poblaciones
2. Búsqueda aleatoria versus Búsqueda local
3. Métodos de búsqueda local básicos
4. Problemas de la búsqueda local

# 1

## Introducción

# Introducción

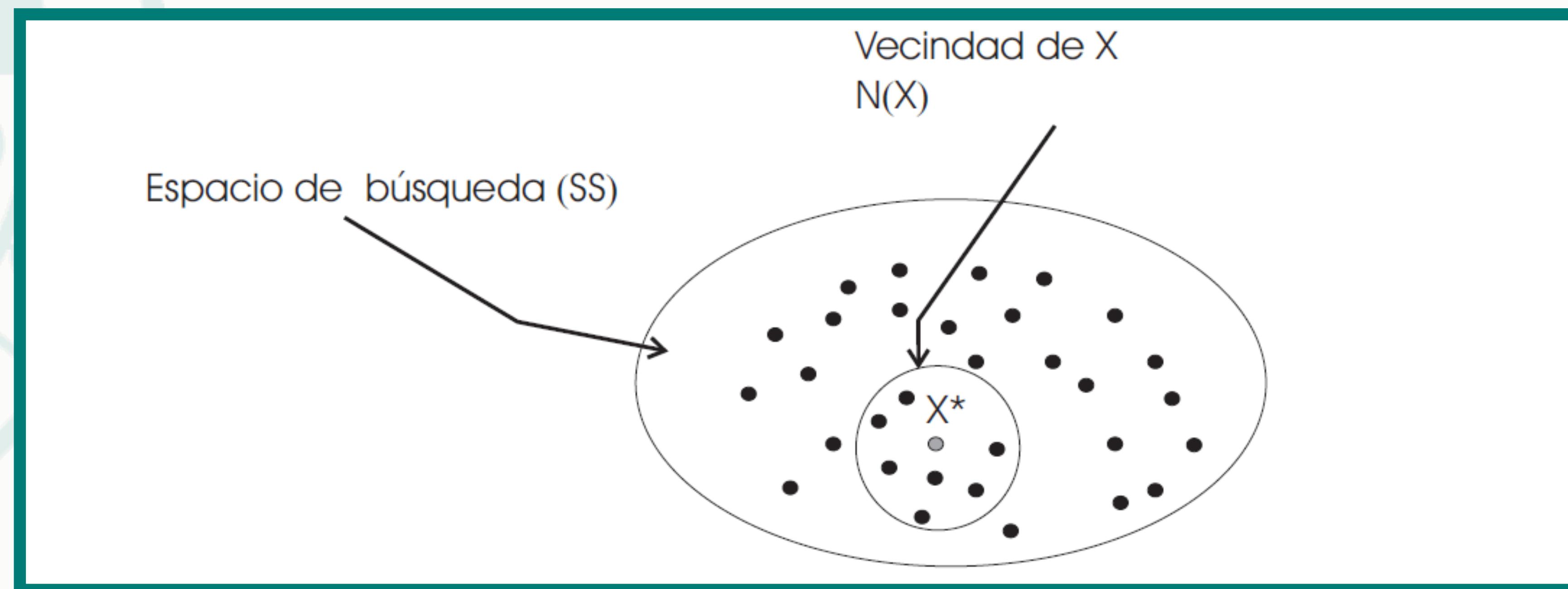
término local

- El término “local” se utiliza frecuentemente en los estudios teóricos y prácticos de las metaheurísticas de búsqueda
- Se asocia al uso de estructuras de entorno, reflejando el concepto de proximidad o vecindad entre las soluciones alternativas del problema

# Introducción

término local

**Todas las soluciones incluidas en el entorno de la solución actual, que viene delimitado por un operador de generación de soluciones, se denominan soluciones vecinas**



# Introducción

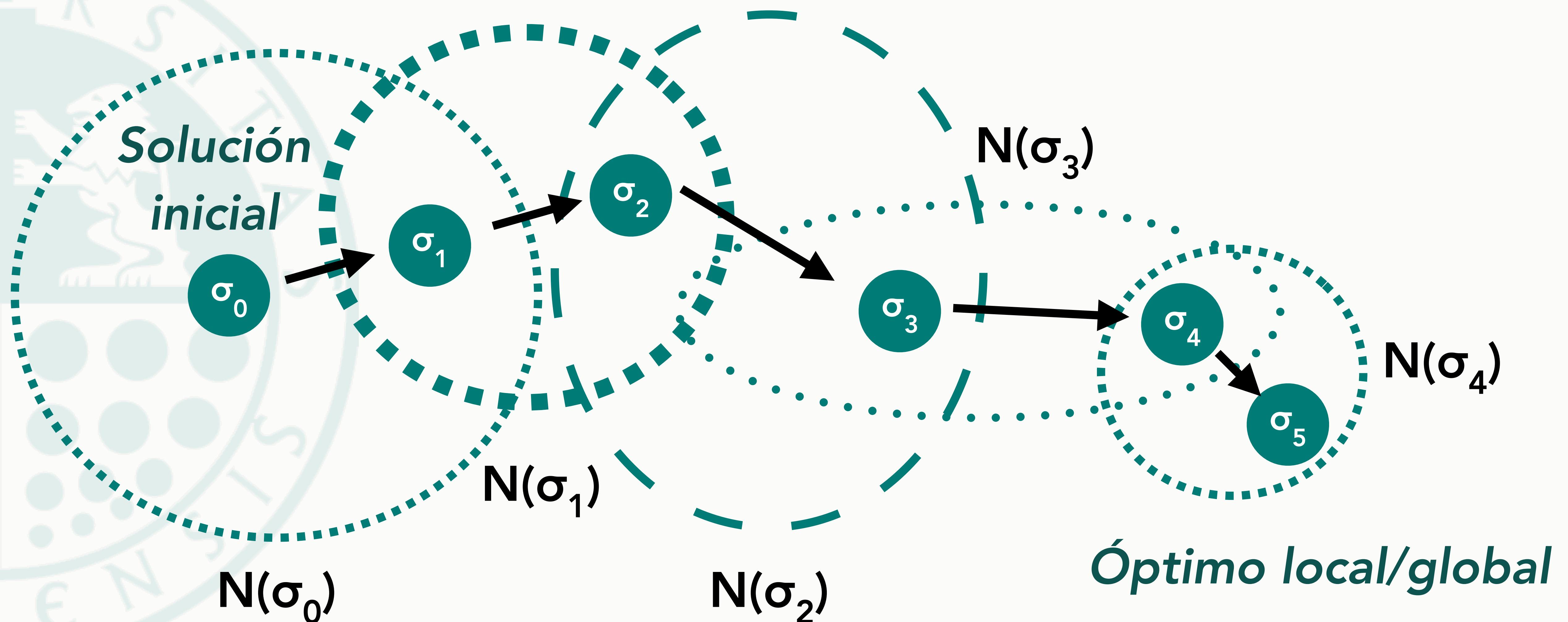
término local

- Los algoritmos basados en esta estrategia efectúan un estudio local del espacio de búsqueda, puesto que analizan el entorno de la solución actual para decidir cómo continuar el recorrido de la búsqueda

Una búsqueda local es un proceso que, dada la solución actual en la que se encuentra el recorrido, selecciona iterativamente una solución de su entorno para continuar la búsqueda

# Introducción

término local



# Introducción

término local

**Una búsqueda local es un proceso que, dada la solución actual en la que se encuentra el recorrido, selecciona iterativamente una solución de su entorno para continuar la búsqueda**

- Basta con diseñar la estructura de entorno para obtener un modelo genérico de algoritmo de búsqueda

# Introducción

término local

**Una búsqueda local es un proceso que, dada la solución actual en la que se encuentra el recorrido, selecciona iterativamente una solución de su entorno para continuar la búsqueda**

## Descripción

- Se fija una codificación para las soluciones
- Se define un operador de generación de vecino y, en consecuencia, se fija una estructura de entorno para las mismas
- Se escoge una solución del entorno de la solución actual hasta que se satisfaga el criterio de parada

# Introducción

elementos básicos en el proceso de búsqueda



# Introducción

elementos básicos en el proceso de búsqueda

## Proceso de elección de la solución inicial

# Introducción

elementos básicos en el proceso de búsqueda

**Proceso de elección de la solución inicial**

**Operador de vecino: Proceso de selección de solución/generación de una solución vecina**

$$S \rightarrow S', S' \in E(S) \equiv N(S)$$

# Introducción

elementos básicos en el proceso de búsqueda

## Proceso de elección de la solución inicial

**Operador de vecino:** Proceso de selección de solución/generación de una solución vecina

$$S \rightarrow S', S' \in E(S) \equiv N(S)$$

Proceso de aceptación de solución vecina como solución actual

# Introducción

## procedimiento búsqueda por entornos

### Inicio

GENERA(Solución Inicial)

Solución Actual  $\leftarrow$  Solución Inicial;

Mejor Solución  $\leftarrow$  Solución Actual;

### Repetir

Solución Vecina  $\leftarrow$  GENERA\_VECINO(Solución Actual);

Si Acepta(Solución Vecina)

entonces Solución Actual  $\leftarrow$  Solución Vecina;

Si Objetivo(Solución Actual) es mejor que Objetivo(Mejor Solución)

entonces Mejor Solución  $\leftarrow$  Solución Actual;

Hasta (Criterio de parada);

DEVOLVER (Mejor Solución);

### Fin

# Introducción

## procedimiento búsqueda por entornos

**Inicio**

**GENERA(Solución Inicial)**

**Solución Actual ← Solución Inicial;**

**Mejor Solución ← Solución Actual;**

**Repetir**

**Solución Vecina ← GENERA\_VECINO(Solución Actual);**

**Si Acepta(Solución Vecina)**

**entonces Solución Actual ← Solución Vecina;**

**Si Objetivo(Solución Actual) es mejor que Objetivo(Mejor Solución)**

**entonces Mejor Solución ← Solución Actual;**

**Hasta (Criterio de parada);**

**DEVOLVER (Mejor Solución);**

**Fin**

# Introducción

## búsqueda basada en poblaciones

En un modelo de poblaciones de soluciones es fundamental definir cómo generar nuevas poblaciones

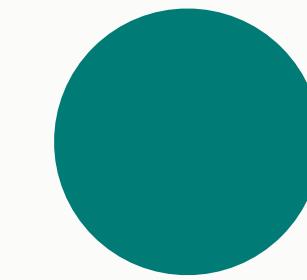
La nueva población se debe asociar a la anterior y considerar la calidad de las anteriores soluciones

# Introducción

## búsqueda basada en poblaciones

En un modelo de poblaciones de soluciones es fundamental definir cómo generar nuevas poblaciones

La nueva población se debe asociar a la anterior y considerar la calidad de las anteriores soluciones

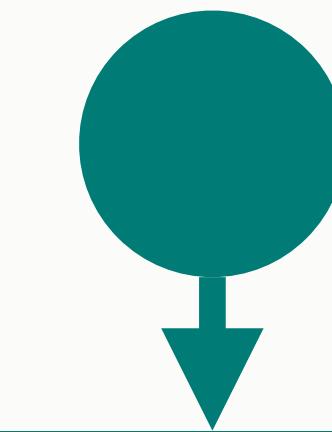


# Introducción

## búsqueda basada en poblaciones

En un modelo de poblaciones de soluciones es fundamental definir cómo generar nuevas poblaciones

La nueva población se debe asociar a la anterior y considerar la calidad de las anteriores soluciones



**Generar población inicial**

# Introducción

## búsqueda basada en poblaciones

En un modelo de poblaciones de soluciones es fundamental definir cómo generar nuevas poblaciones

La nueva población se debe asociar a la anterior y considerar la calidad de las anteriores soluciones



# Introducción

## búsqueda basada en poblaciones

En un modelo de poblaciones de soluciones es fundamental definir cómo generar nuevas poblaciones

La nueva población se debe asociar a la anterior y considerar la calidad de las anteriores soluciones

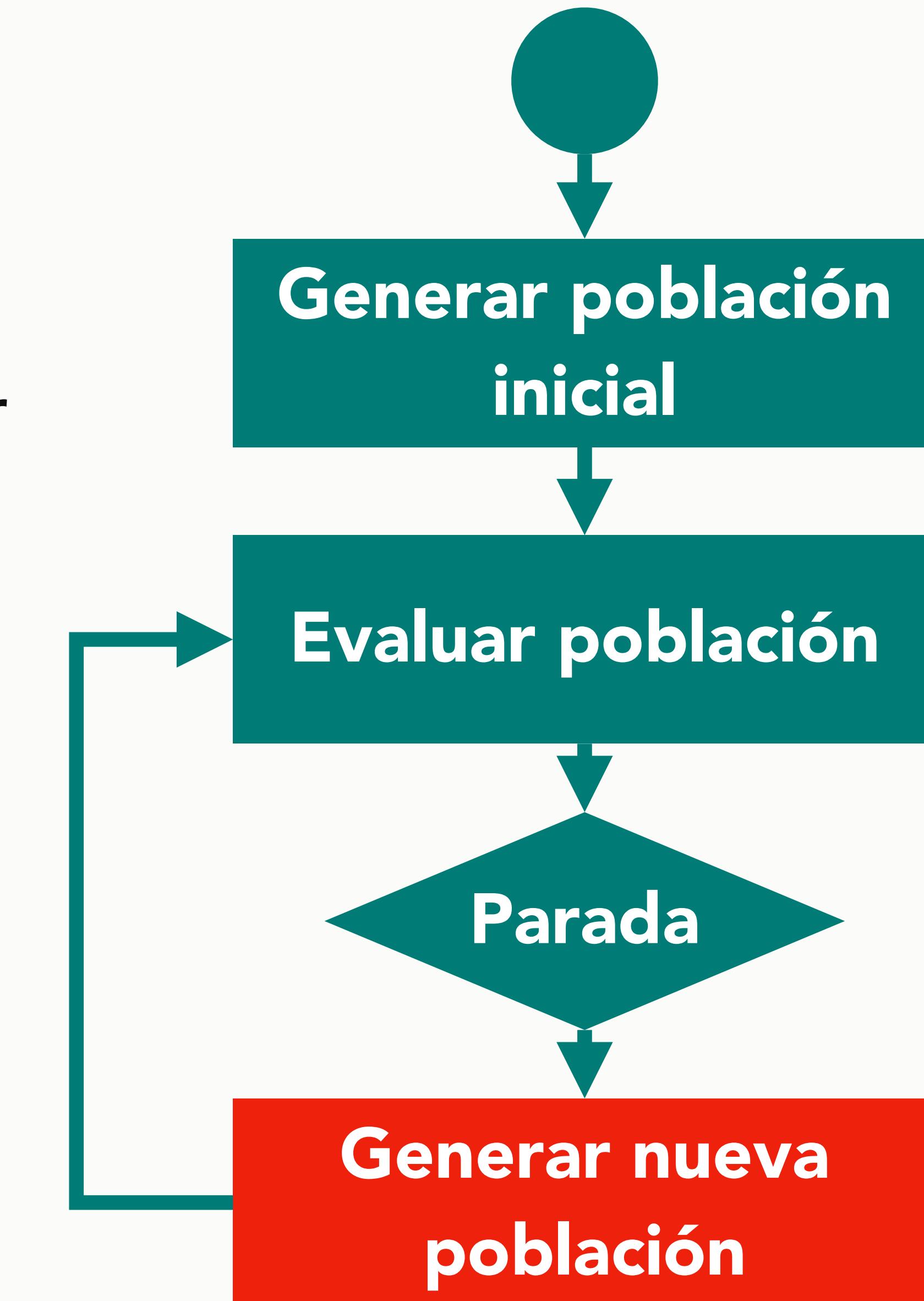


# Introducción

## búsqueda basada en poblaciones

En un modelo de poblaciones de soluciones es fundamental definir cómo generar nuevas poblaciones

La nueva población se debe asociar a la anterior y considerar la calidad de las anteriores soluciones

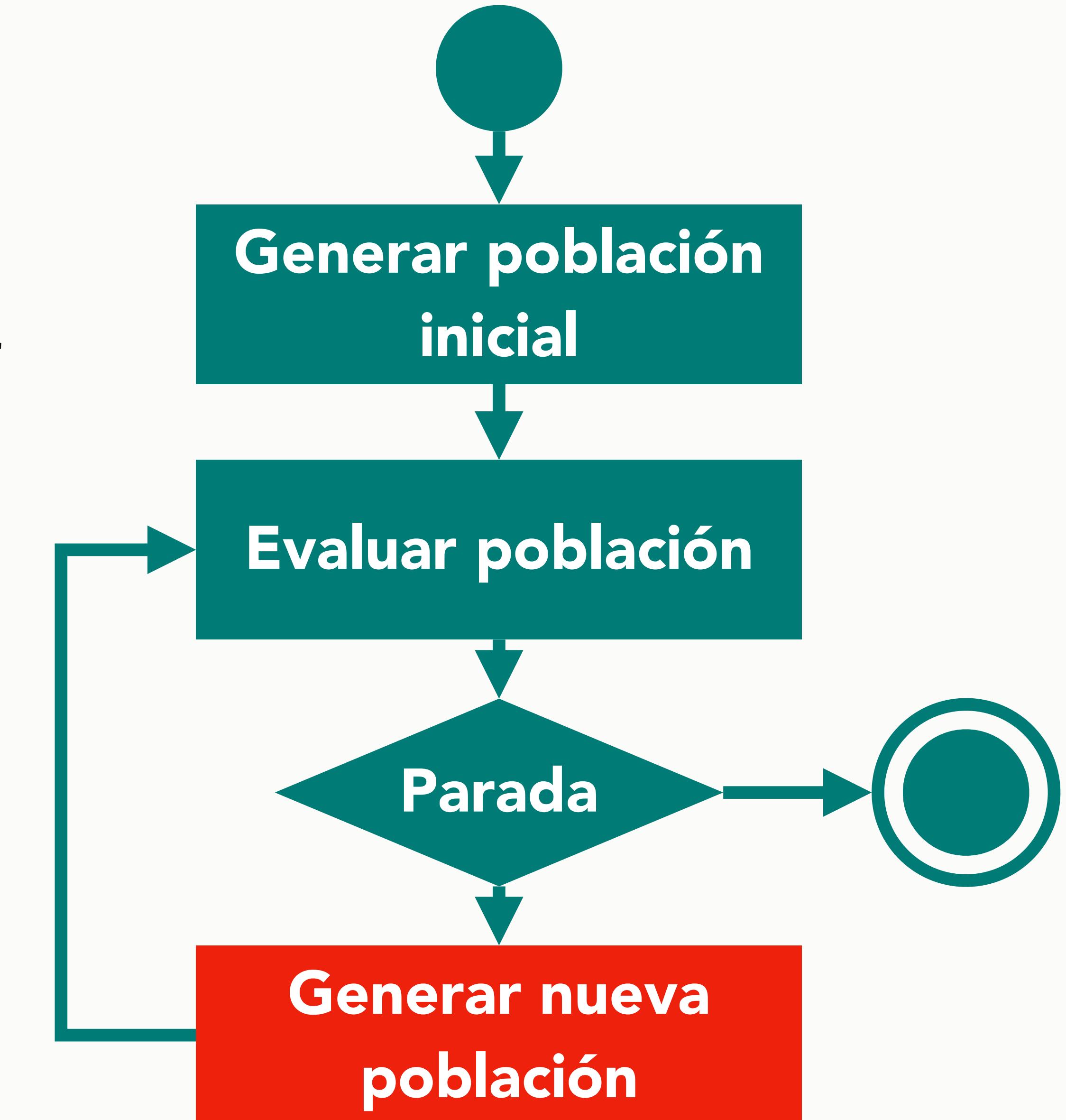


# Introducción

## búsqueda basada en poblaciones

En un modelo de poblaciones de soluciones es fundamental definir cómo generar nuevas poblaciones

La nueva población se debe asociar a la anterior y considerar la calidad de las anteriores soluciones

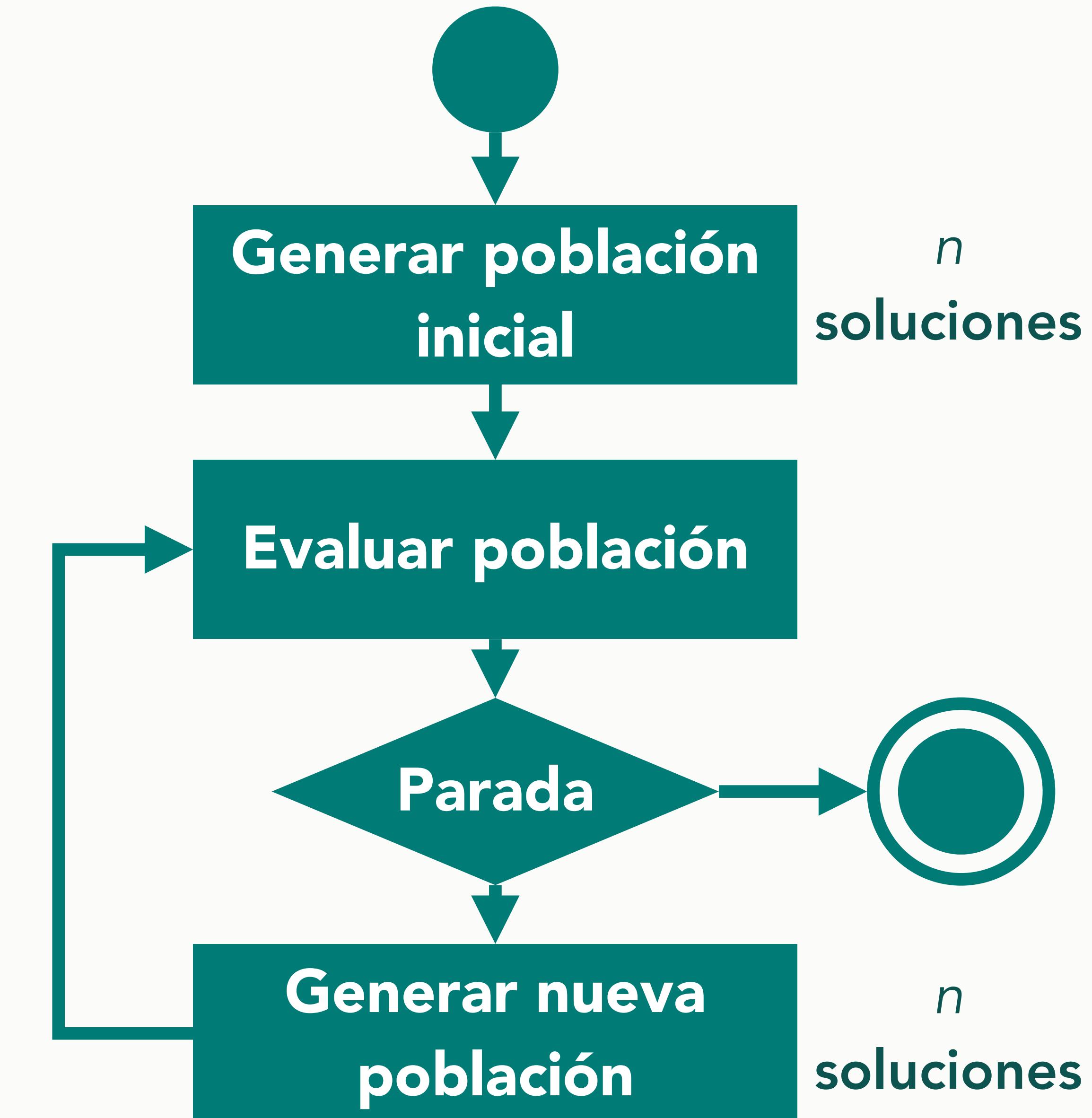


# Introducción

## búsqueda basada en poblaciones

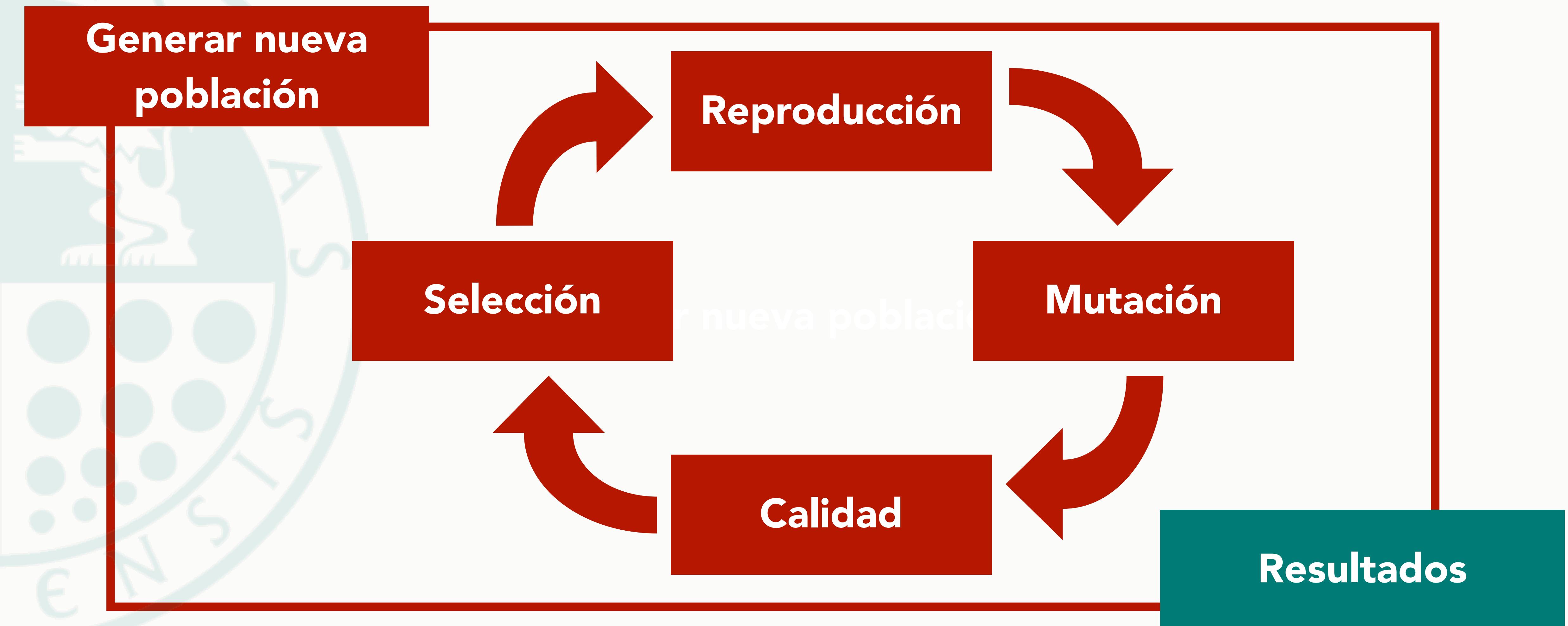
Se podría tener un modelo que seleccione soluciones, opere con ellas y puedan ser reinsertadas en la población dando lugar a una nueva población

Por ejemplo, imitando a la GENÉTICA, combinando cromosomas



# Introducción

## búsqueda basada en poblaciones



# Introducción

## búsqueda basada en poblaciones

Existen otros paradigmas que se pueden utilizar para la obtención de poblaciones:

- Optimización basada en manadas de lobos
- Pájaro cuco
- Colonias de hormigas
- Bandadas de aves
- Colonias de abejas

# 2

## Búsqueda aleatoria VERSUS Búsqueda local

# Búsqueda aleatoria versus Búsqueda local

aleatoria pura

Vamos ahora a analizar el comportamiento de la búsqueda aleatoria, realizando un estudio de su eficacia y eficiencia

Así justificamos el uso de los procedimientos de búsqueda local

# Búsqueda aleatoria versus Búsqueda local

búsqueda aleatoria

Se elige aleatoriamente una muestra de soluciones del espacio de búsqueda y se devuelve la mejor



Se diría que el entorno de una solución es todo el espacio de búsqueda

$E(s) = \text{"todo el espacio de búsqueda"}$

# Búsqueda aleatoria versus Búsqueda local

procedimiento de búsqueda aleatoria

**Inicio**

**GENERA(Solución Inicial)**

**Solución Actual  $\leftarrow$  Solución Inicial;**

**Mejor Solución  $\leftarrow$  Solución Actual;**

**Repetir**

**GENERA(Solución Actual);**

**Si Objetivo(Solución Actual) es mejor que Objetivo(Mejor Solución)**

**entonces Mejor Solución  $\leftarrow$  Solución Actual;**

**Hasta (Criterio de parada);**

**DEVOLVER (Mejor Solución);**

**Fin**

**Generación Aleatoria**

# Búsqueda aleatoria versus Búsqueda local

búsqueda aleatoria en el viajante de comercio

| Iteración | Solución            |
|-----------|---------------------|
| 1         | (1 2 4 3 8 5 9 6 7) |
| 2         | (9 6 4 7 8 5 1 2 3) |
| 3         | (2 4 1 5 8 3 2 6 9) |

# Búsqueda aleatoria versus Búsqueda local

búsqueda aleatoria en el viajante de comercio

| Iteración | Solución            |
|-----------|---------------------|
| 1         | (1 2 4 3 8 5 9 6 7) |
| 2         | (9 6 4 7 8 5 1 2 3) |
| 3         | (2 4 1 5 8 3 2 6 9) |
| 4         | (7 6 9 5 8 3 1 2 4) |

# Búsqueda aleatoria versus Búsqueda local

búsqueda aleatoria en el viajante de comercio

| Iteración | Solución            |
|-----------|---------------------|
| 1         | (1 2 4 3 8 5 9 6 7) |
| 2         | (9 6 4 7 8 5 1 2 3) |
| 3         | (2 4 1 5 8 3 2 6 9) |
| 4         | (7 6 9 5 8 3 1 2 4) |
| 5         | (8 3 7 2 1 5 7 6 9) |

# Búsqueda aleatoria versus Búsqueda local

búsqueda aleatoria en el viajante de comercio

| Iteración | Solución            |
|-----------|---------------------|
| 1         | (1 2 4 3 8 5 9 6 7) |
| 2         | (9 6 4 7 8 5 1 2 3) |
| 3         | (2 4 1 5 8 3 2 6 9) |
| 4         | (7 6 9 5 8 3 1 2 4) |
| 5         | (8 3 7 2 1 5 7 6 9) |
| 6         | (2 5 1 3 9 8 4 6 7) |

# Búsqueda aleatoria versus Búsqueda local

búsqueda aleatoria en el viajante de comercio

| Iteración | Solución            |
|-----------|---------------------|
| 1         | (1 2 4 3 8 5 9 6 7) |
| 2         | (9 6 4 7 8 5 1 2 3) |
| 3         | (2 4 1 5 8 3 2 6 9) |
| 4         | (7 6 9 5 8 3 1 2 4) |
| 5         | (8 3 7 2 1 5 7 6 9) |
| 6         | (2 5 1 3 9 8 4 6 7) |
| 7         | (4 7 5 1 8 3 2 6 9) |

# Búsqueda aleatoria versus Búsqueda local

búsqueda aleatoria en el viajante de comercio

| Iteración | Solución            |
|-----------|---------------------|
| 1         | (1 2 4 3 8 5 9 6 7) |
| 2         | (9 6 4 7 8 5 1 2 3) |
| 3         | (2 4 1 5 8 3 2 6 9) |
| 4         | (7 6 9 5 8 3 1 2 4) |
| 5         | (8 3 7 2 1 5 7 6 9) |
| 6         | (2 5 1 3 9 8 4 6 7) |
| 7         | (4 7 5 1 8 3 2 6 9) |
| 8         | (1 4 2 3 8 5 6 9 7) |

# Búsqueda aleatoria versus Búsqueda local

estudio teórico de la eficiencia de la búsqueda aleatoria

Si el problema tiene  $m$  soluciones y el óptimo es único, la probabilidad de que al generar aleatoriamente una solución se obtenga la óptima es  $1/m$

Sean  $A_i$  los sucesos siguientes, que determinan la probabilidad absoluta de obtener el óptimo en la iteración  $i$ :

$$P(A_i) = \frac{1}{m} \quad \forall i = 1, \dots, n$$

# Búsqueda aleatoria versus Búsqueda local

estudio teórico de la eficiencia de la búsqueda aleatoria

La probabilidad de obtener el óptimo en n iteraciones sería:

$$= 1 - P((A_1 \cup A_2 \cup \dots \cup A_n)^c)$$

$$= 1 - P(A_1^c \cap A_2^c \cap \dots \cap A_n^c)$$

$$= 1 - P(A_1^c) P(A_2^c) \dots P(A_n^c)$$

$$= 1 - (1 - P(A_1)) (1 - P(A_2)) \dots (1 - P(A_n))$$

$$= 1 - (1 - \frac{1}{m})(1 - \frac{1}{m}) \dots (1 - \frac{1}{m})$$

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = 1 - (1 - \frac{1}{m})^n > 1 - \alpha$$

**La expresión anterior se puede emplear para derivar el valor de n que, con una probabilidad suficiente grande, garantiza la eficacia del método**

# Búsqueda aleatoria versus Búsqueda local

estudio teórico de la eficiencia de la búsqueda aleatoria

La probabilidad de obtener el óptimo en n iteraciones sería:

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = 1 - (1 - \frac{1}{m})^n > 1 - \alpha$$

$$\alpha > (1 - \frac{1}{m})^n$$

**Si  $\alpha$  es tal que  $0 < \alpha < 1$  (probabilidad a priori de error en la búsqueda, es decir, probabilidad de que la búsqueda aleatoria no encuentre el óptimo).**

**n** sería el nº de iteraciones necesario para garantizar la obtención del óptimo con probabilidad  $1-\alpha$

$$n > \frac{\log(\alpha)}{\log(1 - 1/m)}$$

# Búsqueda aleatoria versus Búsqueda local

estudio teórico de la eficiencia de la búsqueda aleatoria

| Prechazo a | Péxito 1-a | Soluciones | Iteraciones |
|------------|------------|------------|-------------|
| 0.1        | 0.9        | 1000       | 2302        |
| 0.2        | 0.8        |            | 1609        |
| 0.3        | 0.7        |            | 1203        |
| 0.4        | 0.6        |            | 916         |
| 0.1        | 0.9        | 2000       | 4605        |
| 0.2        | 0.8        |            | 3219        |
| 0.3        | 0.7        |            | 2408        |
| 0.4        | 0.6        |            | 1833        |
| 0.1        | 0.9        | 3000       | 6907        |
| 0.2        | 0.8        |            | 4828        |
| 0.3        | 0.7        |            | 3612        |
| 0.4        | 0.6        |            | 2748        |

- n iteraciones
- M tamaño del espacio de búsqueda
- a probabilidad de fallo

# Búsqueda aleatoria versus Búsqueda local

estudio teórico de la eficiencia de la búsqueda aleatoria

| Prechazo a | Péxito 1-a | Soluciones | Iteraciones |
|------------|------------|------------|-------------|
| 0.1        | 0.9        |            | 2302        |
| 0.2        | 0.8        |            | 1609        |
| 0.3        | 0.7        |            | 1203        |
| 0.4        | 0.6        |            | 916         |
| 0.1        | 0.8        | 1000       | 4605        |
| 0.4        | 0.6        |            | 2748        |

**Recordamos que el problema del viajante de comercio tiene un  $O(n!)$**

**Tamaño 7 >>  $7! = 5040$  soluciones**

- n iteraciones
- M tamaño del espacio de búsqueda
- a probabilidad de fallo

# Búsqueda aleatoria versus Búsqueda local

búsqueda aleatoria por recorrido al azar

- Se obtiene desde la descripción de una Búsqueda por Entornos
- La solución inicial se genera aleatoriamente
- El entorno de cualquier solución es propio
- La solución vecina a la solución actual se escoge aleatoriamente dentro del entorno y se acepta automáticamente
- Se almacena la mejor solución obtenida hasta el momento
- En definitiva, puede considerarse como una búsqueda local en la que se acepta el primer vecino generado, independientemente de que sea mejor o peor que la solución actual

# 3

## Métodos de búsqueda local básicos

# Métodos de búsqueda local básicos

## introducción

Consiste en el muestreo de soluciones vecinas mejores que la actual en el entorno de ésta

Hay dos versiones:

- del Mejor
- del Primer Mejor

En ambos casos, el algoritmo devuelve la última solución visitada  
(no es necesario ir almacenando la mejor solución)

# Métodos de búsqueda local básicos

procedimiento general búsqueda local

**Inicio**

  Genera(Solución Inicial);

  Solución Actual  $\leftarrow$  Solución Inicial;

**Repetir**

  Genera\_Solución\_Entorno(Solución Vecina tal que  
    Objetivo(Solución Vecina) **mejor que** Objetivo(Solución Actual));

**Si** Objetivo(Solución Vecina) **mejor que** Objetivo(Solución Actual)

**entonces** Solución Actual  $\leftarrow$  Solución Vecina;

**Hasta** (Objetivo(Solución Vecina) **peor o igual que**

    Objetivo(Solución Actual),  $\forall S \in E(\text{Solución Actual})$ );

DEVOLVER(Solución Actual);

**Fin**

**Generación por Entornos**

**¿y el Mejor?**

# Métodos de búsqueda local básicos

búsqueda local del mejor

## Steepest-Ascent Hill Climbing (best neighbour)

- Genera el entorno completo de la solución actual y selecciona la mejor solución vecina
- Si ésta es mejor que la solución actual, la sustituye y se continúa la iteración
- En otro caso, el algoritmo finaliza

# Métodos de búsqueda local básicos

búsqueda local del mejor

**Inicio**

**GENERA( $S_{act}$ );**

**Repetir**

    Mejor Vecino  $\leftarrow S_{act}$

**Repetir**  $\forall S \in E(S_{act})$

$S' \leftarrow \text{GENERA\_VECINO}(S_{act});$

**Si** Objetivo( $S'$ ) **mejor que** Objetivo(Mejor Vecino) **entonces**

        Mejor Vecino  $\leftarrow S';$

**Fin-Repetir-para**

**Si** Objetivo(Mejor Vecino) **mejor que** Objetivo( $S_{act}$ ) **entonces**

$S_{act} \leftarrow \text{Mejor Vecino};$

**Hasta** (Objetivo(Mejor Vecino) **peor o igual** que Objetivo( $S_{act}$ ));

**DEVOLVER( $S_{act}$ );**

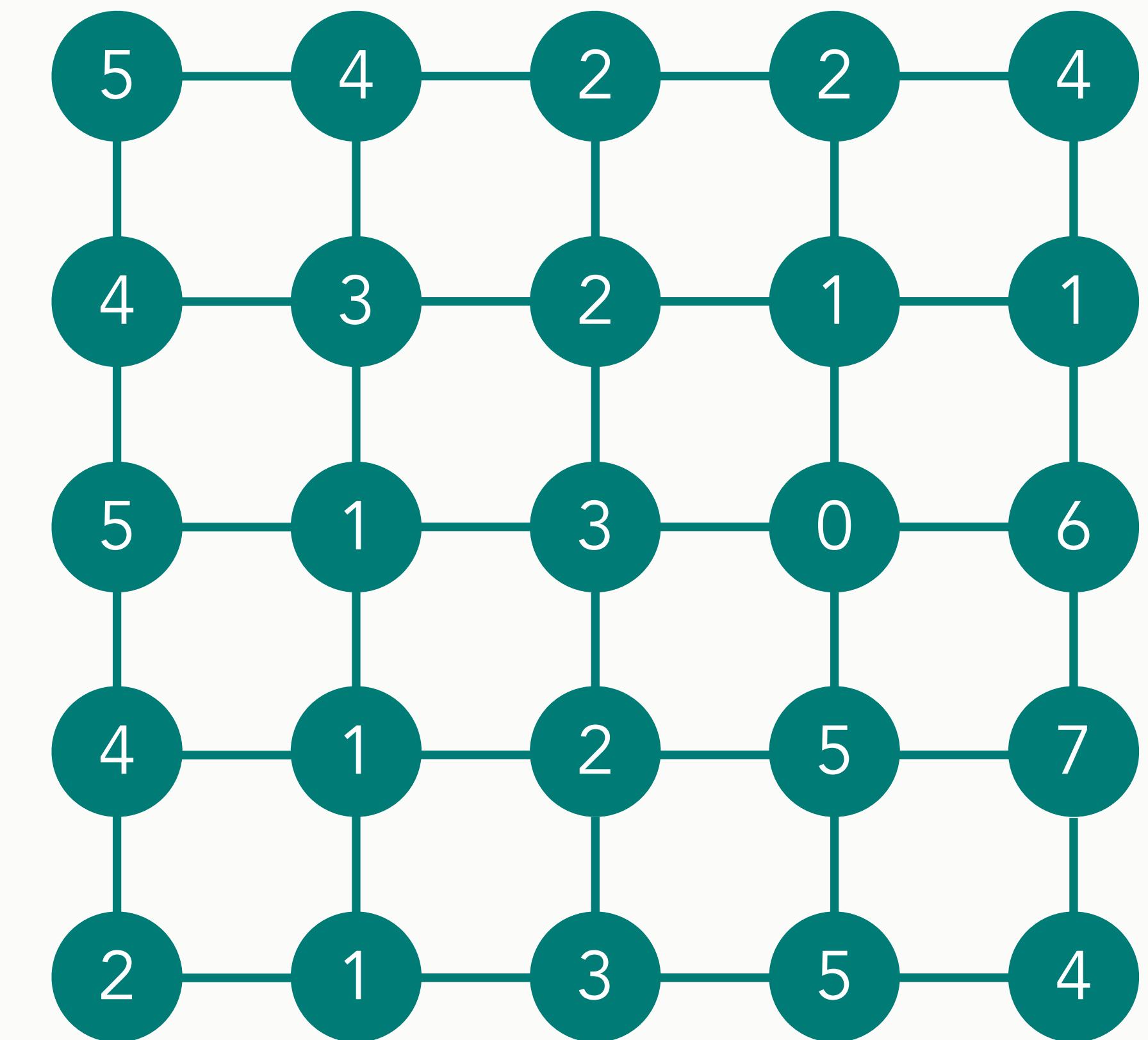
**Fin**

# Métodos de búsqueda local básicos

búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$



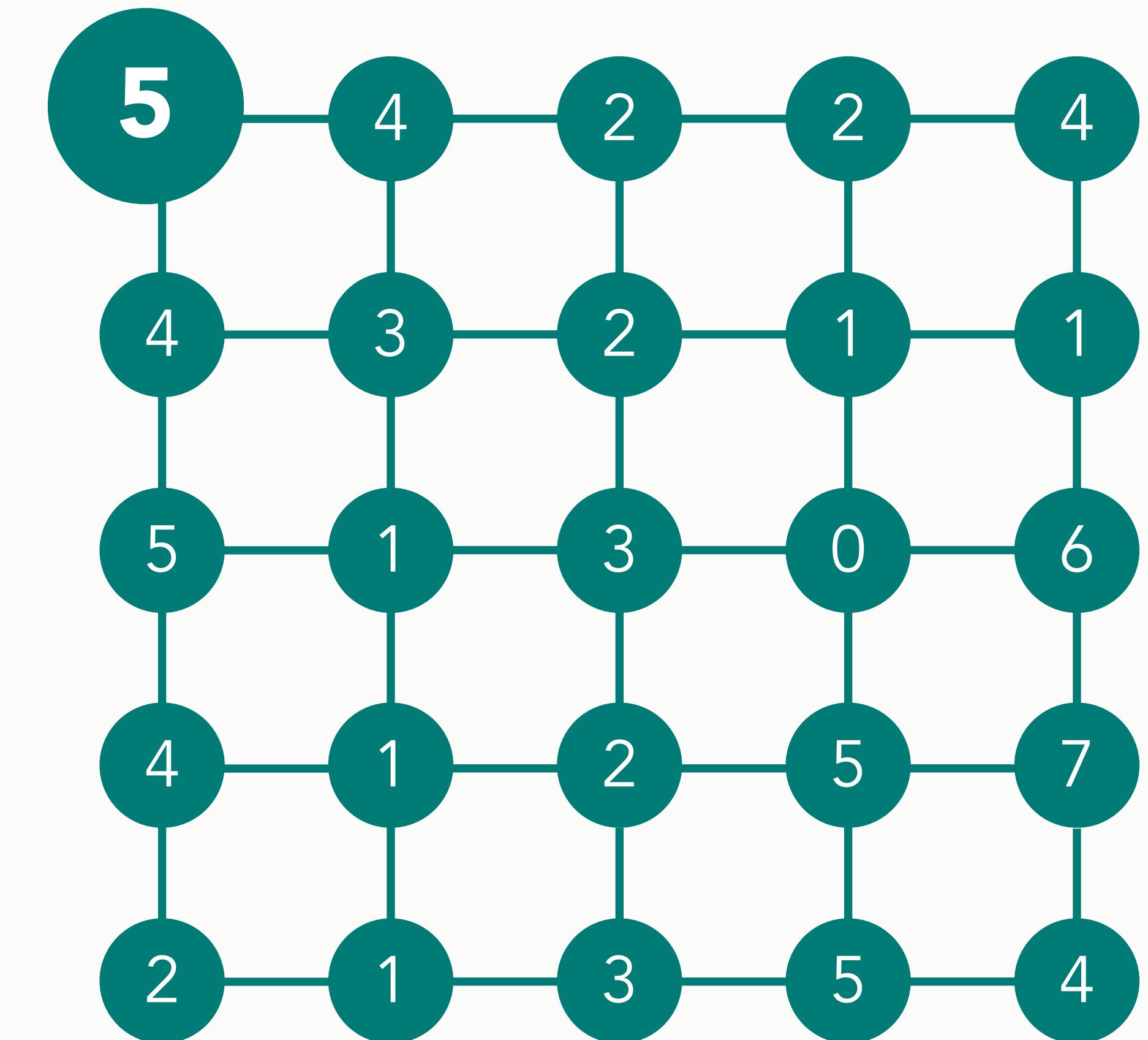
# Métodos de búsqueda local básicos

búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial



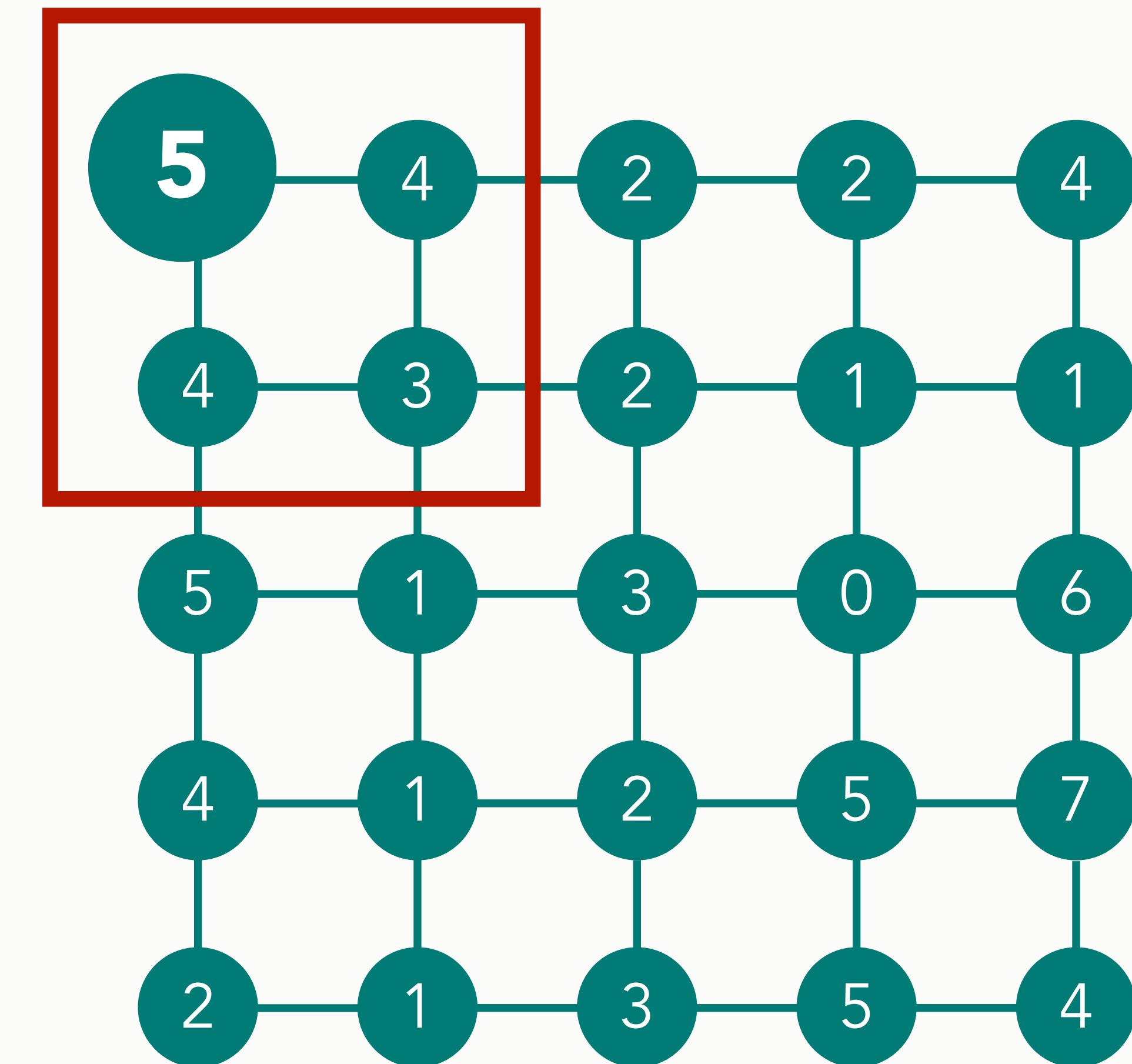
# Métodos de búsqueda local básicos

búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial



# Métodos de búsqueda local básicos

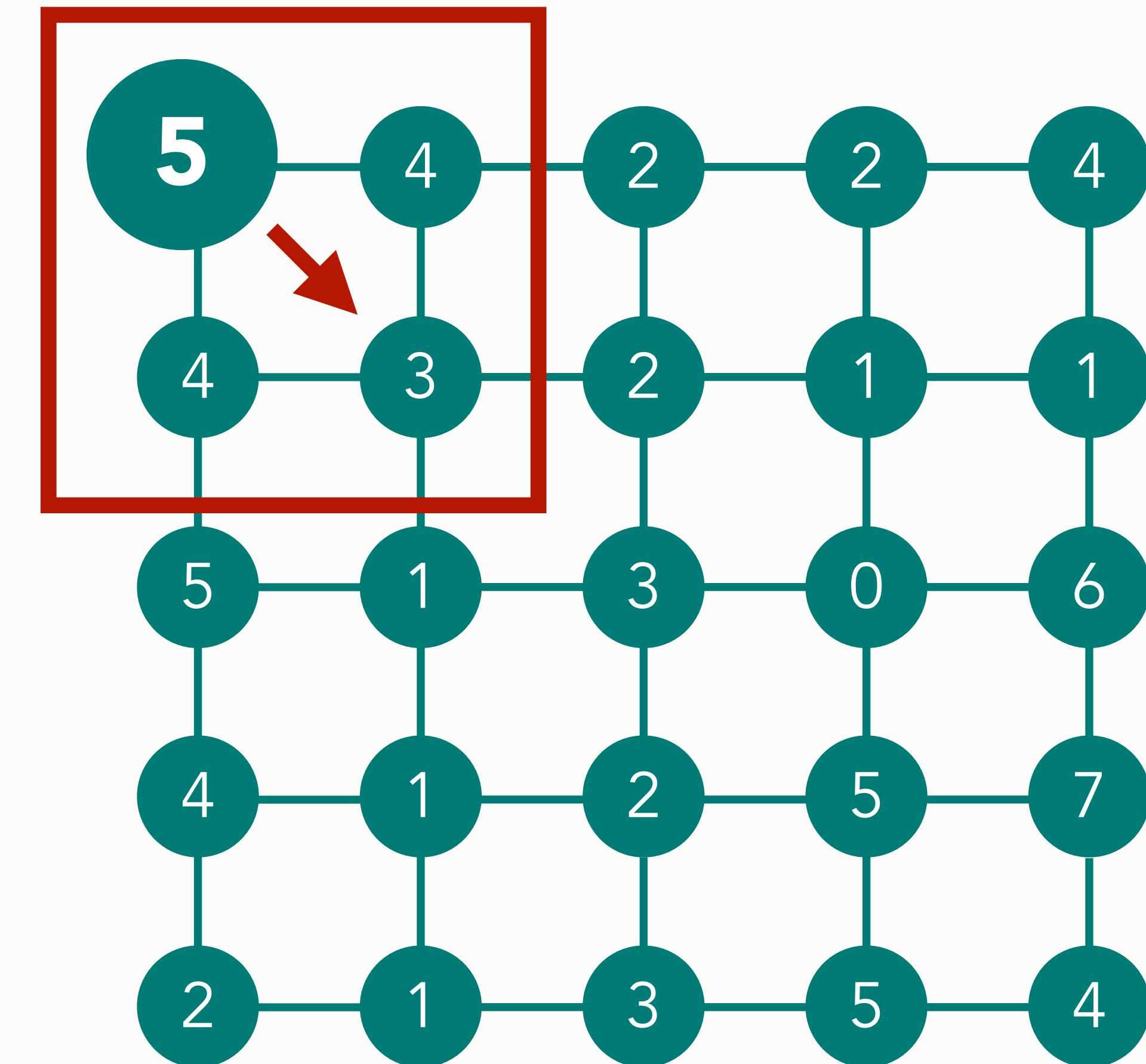
búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local  
del mejor



# Métodos de búsqueda local básicos

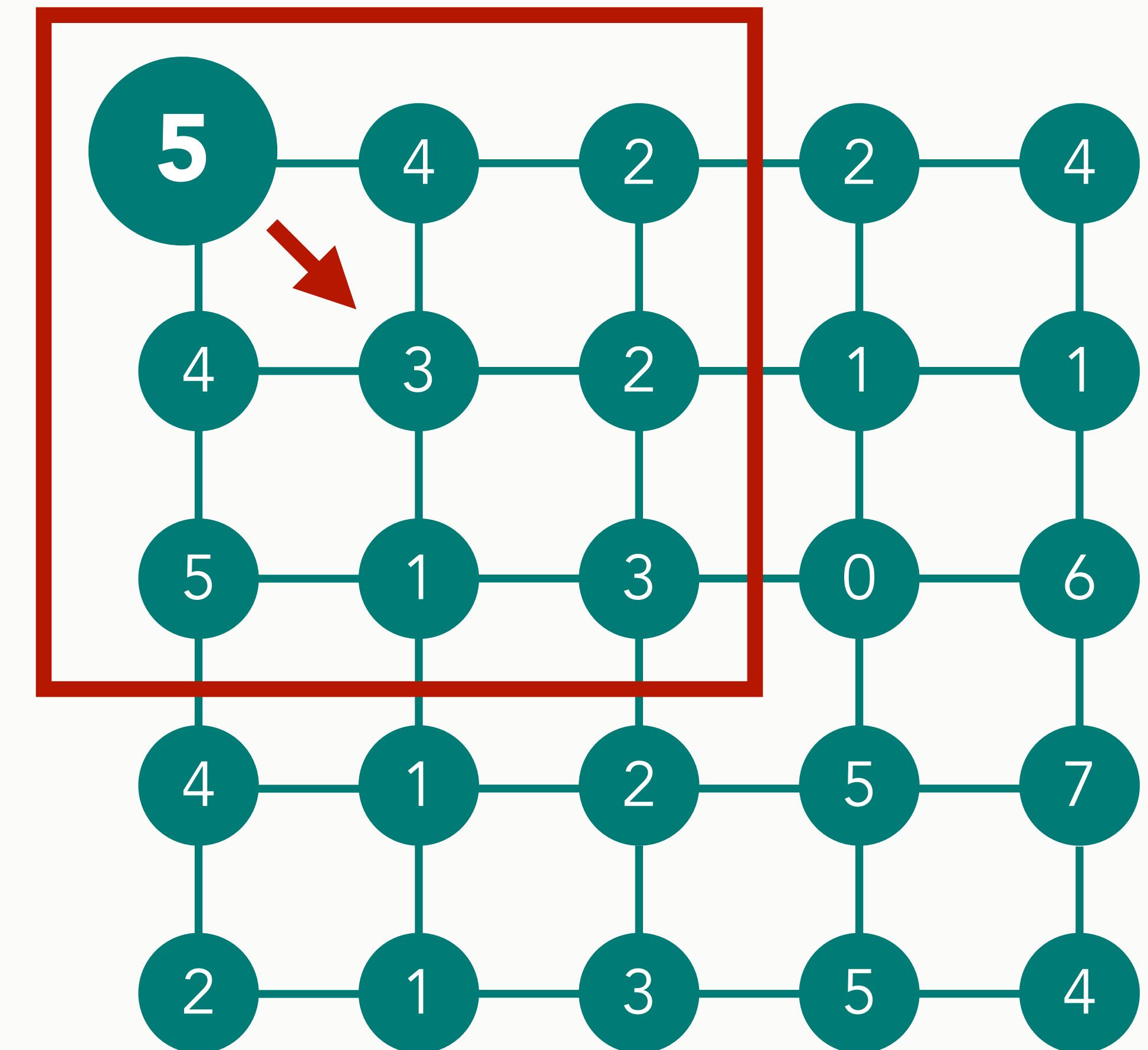
búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local  
del mejor



# Métodos de búsqueda local básicos

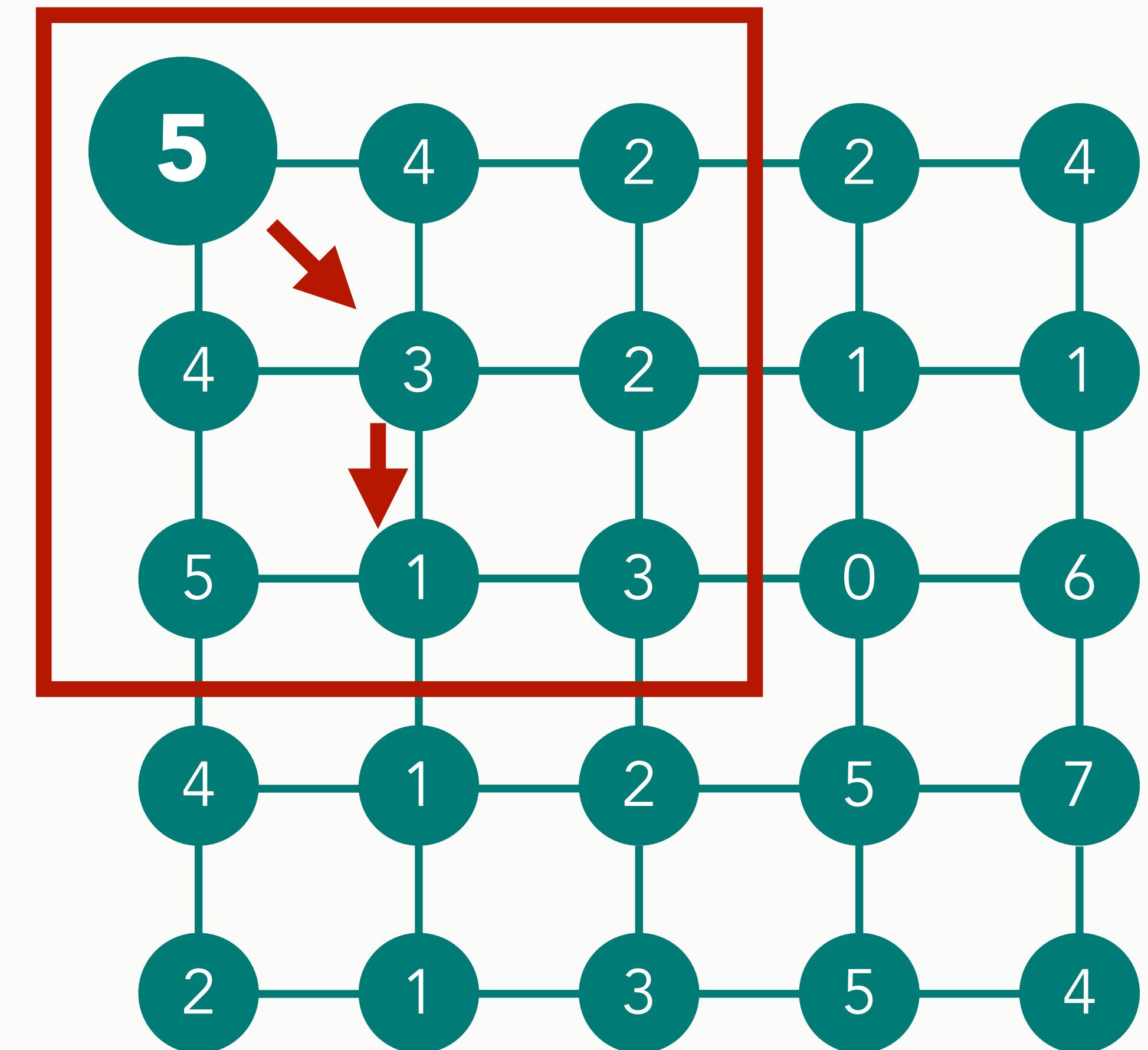
búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local  
del mejor



# Métodos de búsqueda local básicos

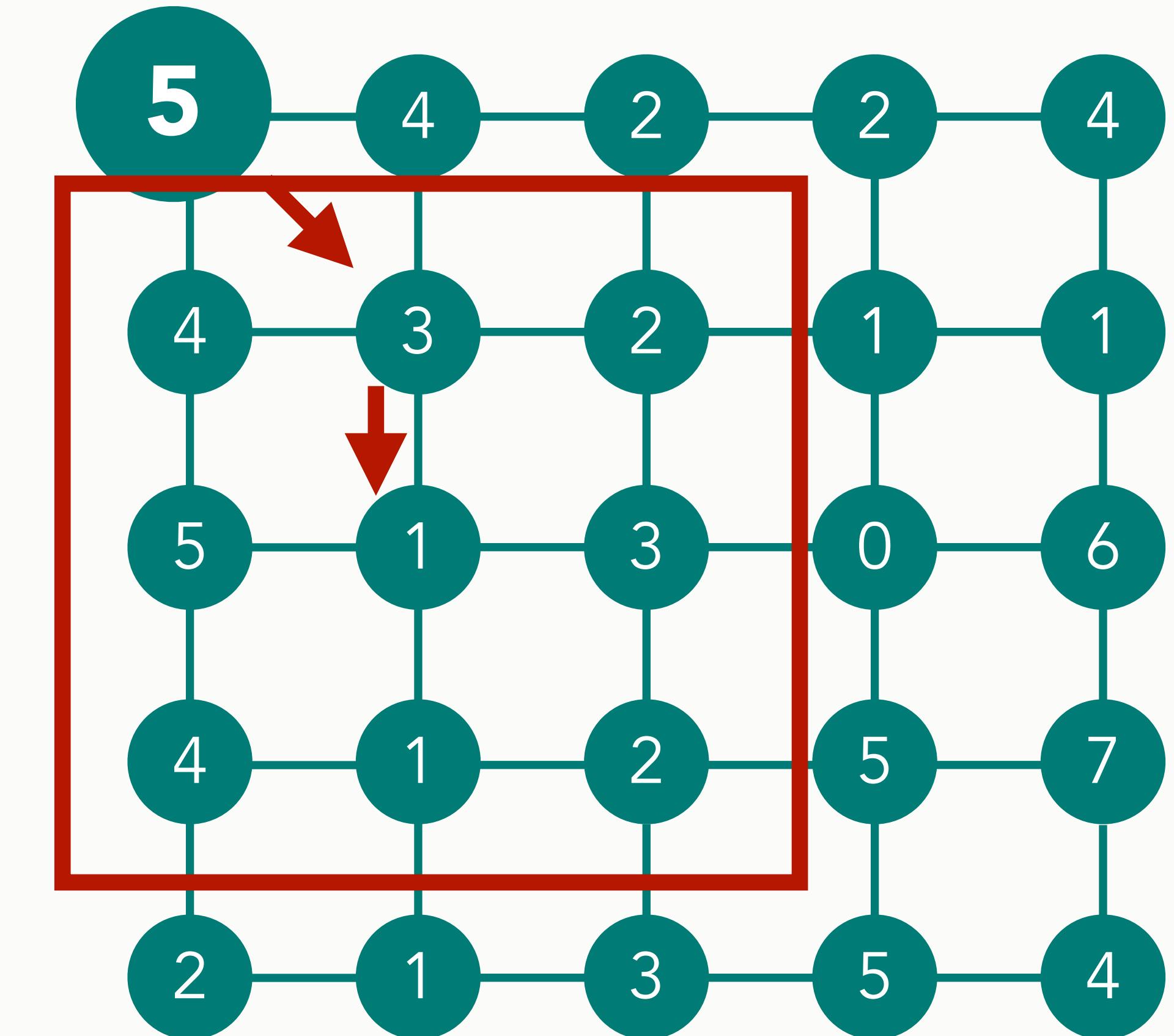
búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local  
del mejor



# Métodos de búsqueda local básicos

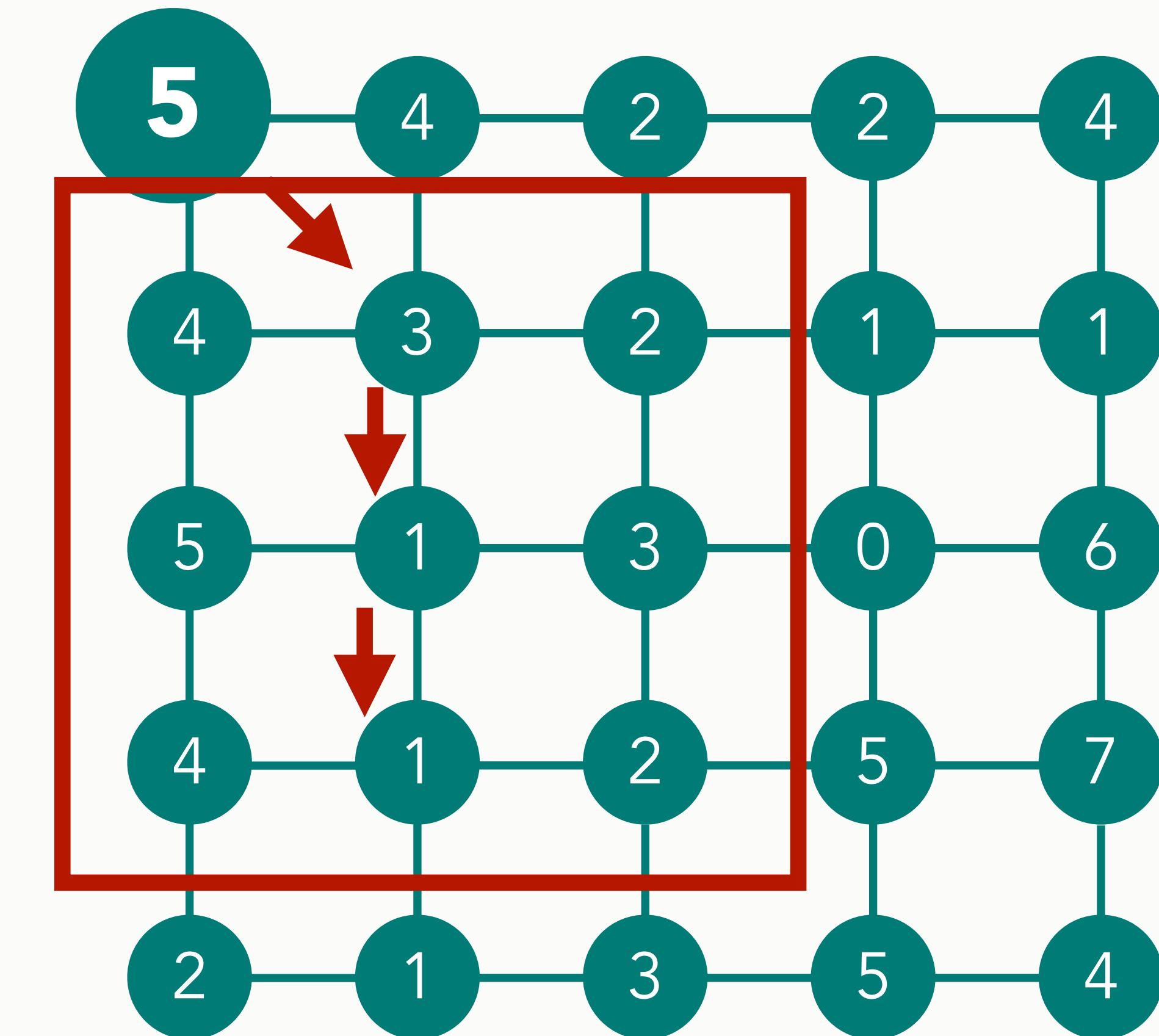
búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local  
del mejor



# Métodos de búsqueda local básicos

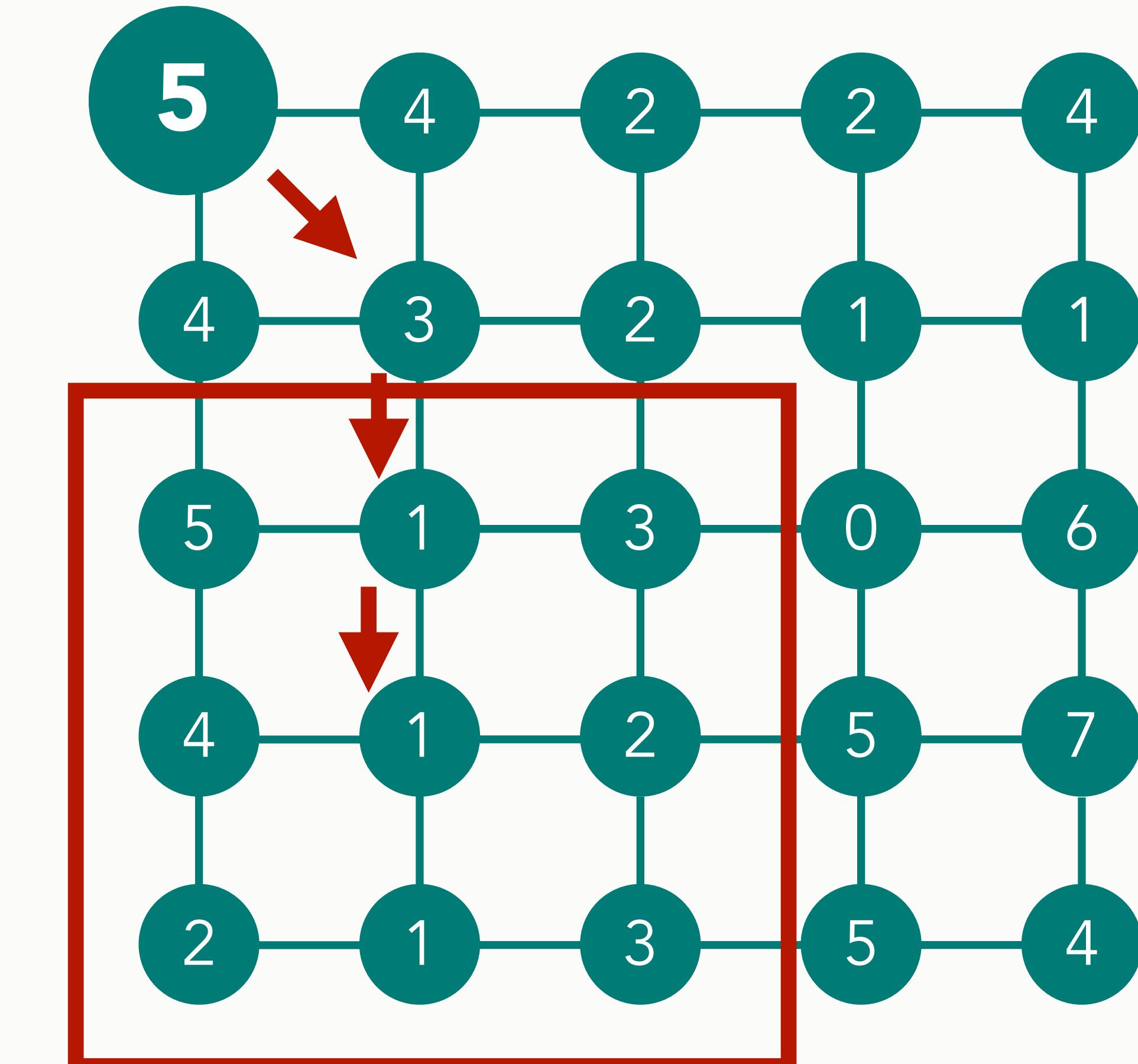
búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local  
del mejor



# Métodos de búsqueda local básicos

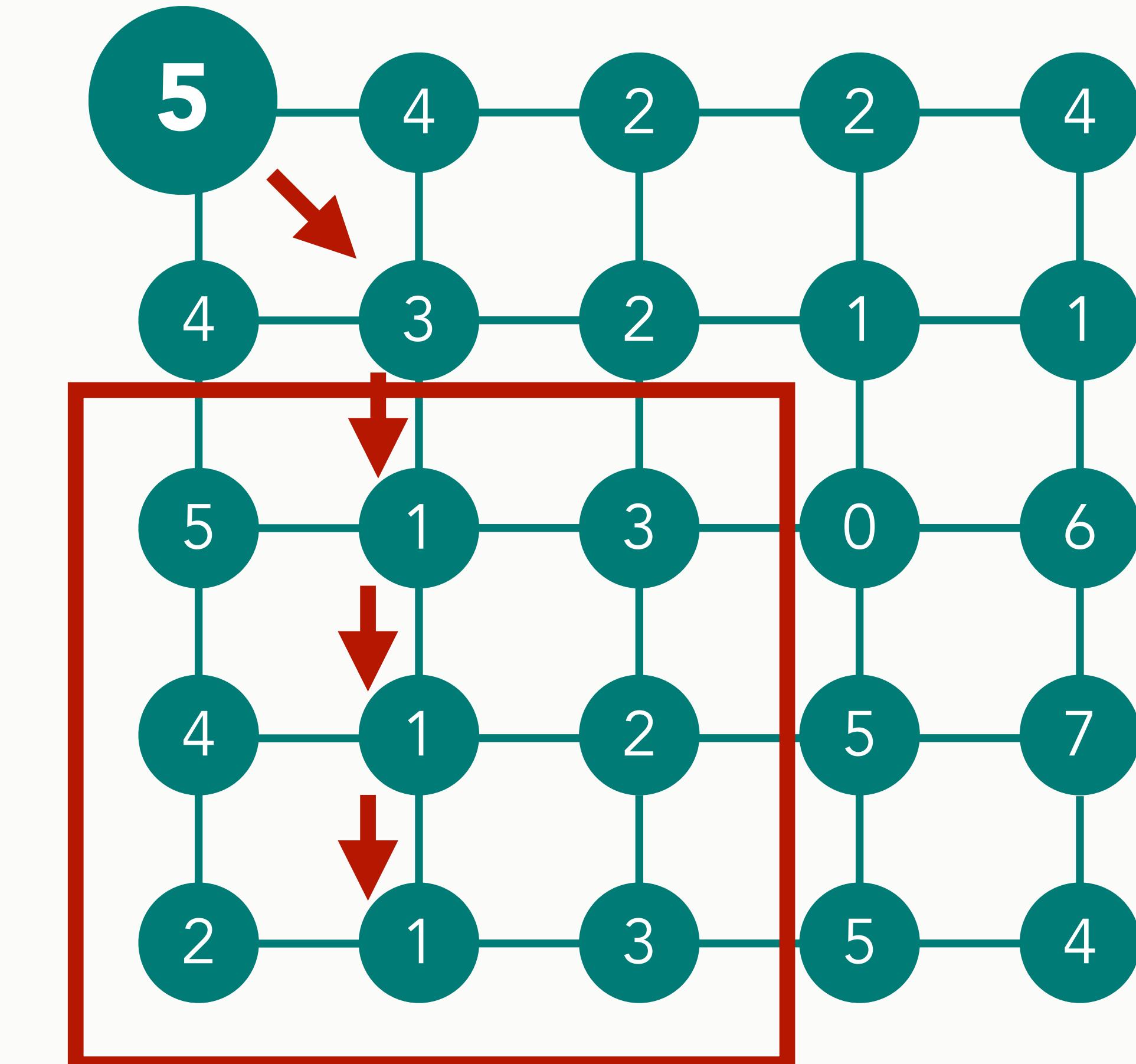
búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local  
del mejor



# Métodos de búsqueda local básicos

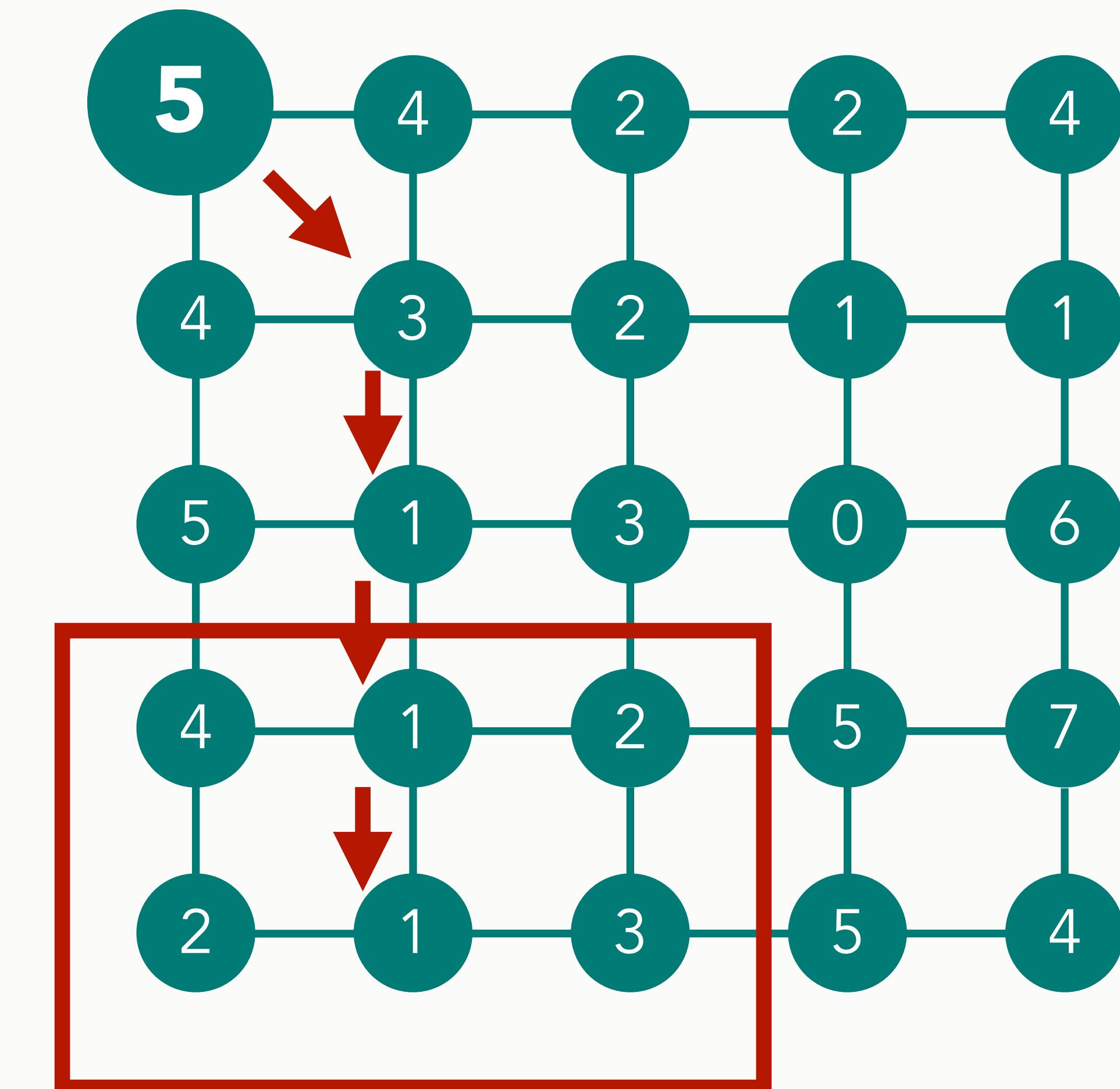
búsqueda local del mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local  
del mejor



# Métodos de búsqueda local básicos

# búsqueda local del mejor

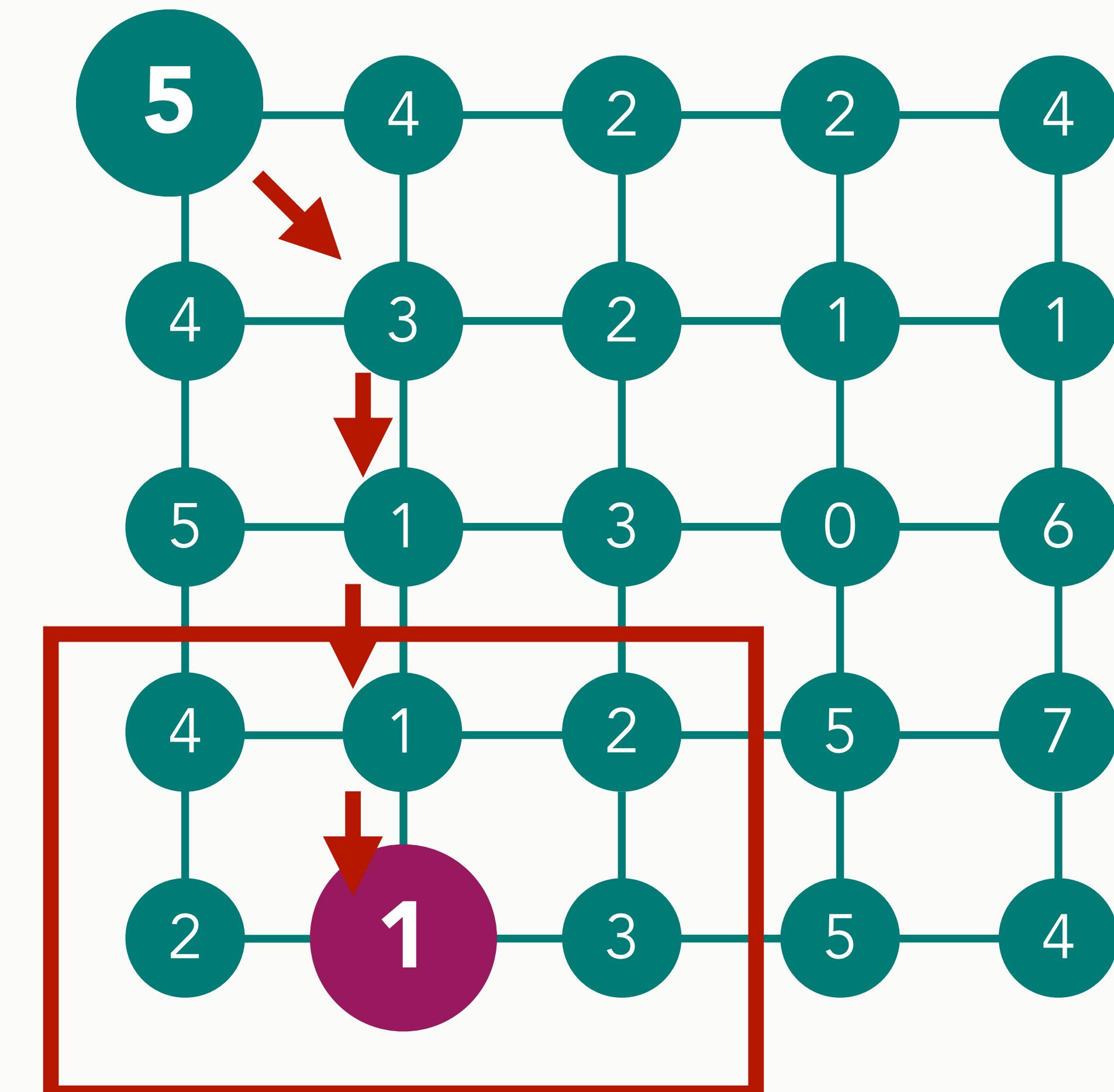
Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_j/S_j = (x_j \pm \{0,1\}, y_j \pm \{0,1\}) \wedge S_j \neq S\}$$

# Solución inicial

# Trayectoria de búsqueda loca del mejor

# Solución devuelta



# Métodos de búsqueda local básicos

búsqueda local del primer mejor

## Simple Hill Climbing (first-best neighbour)

- Se va generando paso a paso el entorno de la solución actual hasta que se obtiene una solución vecina que mejora a la actual o se construye el entorno completo
- En el primer caso, la solución vecina sustituye a la actual y se continúa iterando
- En el segundo, se finaliza la ejecución del algoritmo

# Métodos de búsqueda local básicos

búsqueda local del primer mejor

**Inicio**

**GENERA( $S_{act}$ );**

**Repetir**

**Repetir**

$S' \leftarrow \text{GENERA\_VECINO}(S_{act});$

**Hasta (Objetivo( $S'$ ) mejor que Objetivo( $S_{act}$ )) (se ha generado  $E(S_{act})$  al completo)**

**Si Objetivo( $S'$ ) mejor que Objetivo( $S_{act}$ ) entonces**

$S_{act} \leftarrow S';$

**Hasta (Objetivo( $S'$ ) peor o igual que Objetivo( $S_{act}$ ));**

**DEVOLVER( $S_{act}$ );**

**Fin**

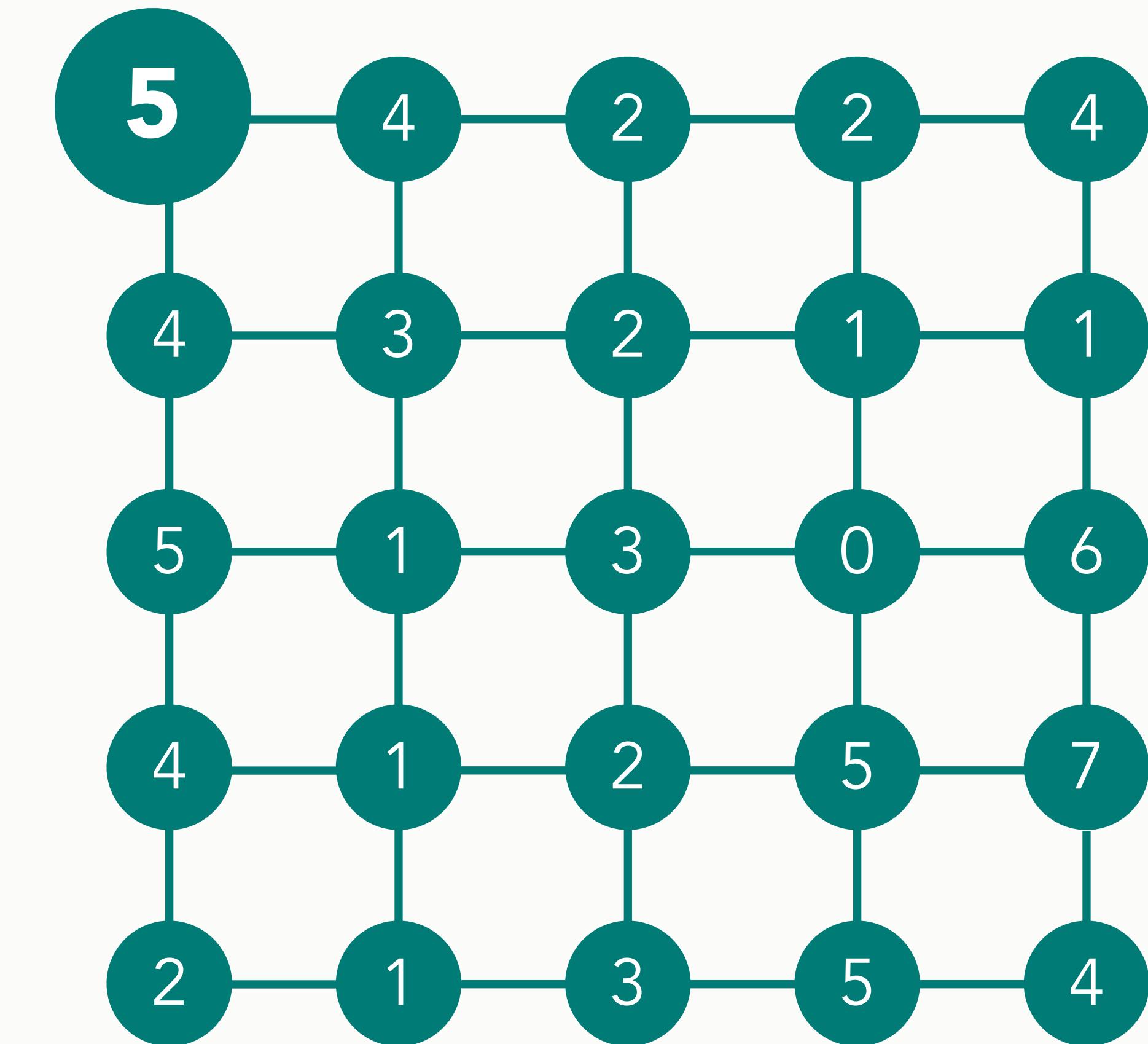
# Métodos de búsqueda local básicos

búsqueda local del primer mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial



# Métodos de búsqueda local básicos

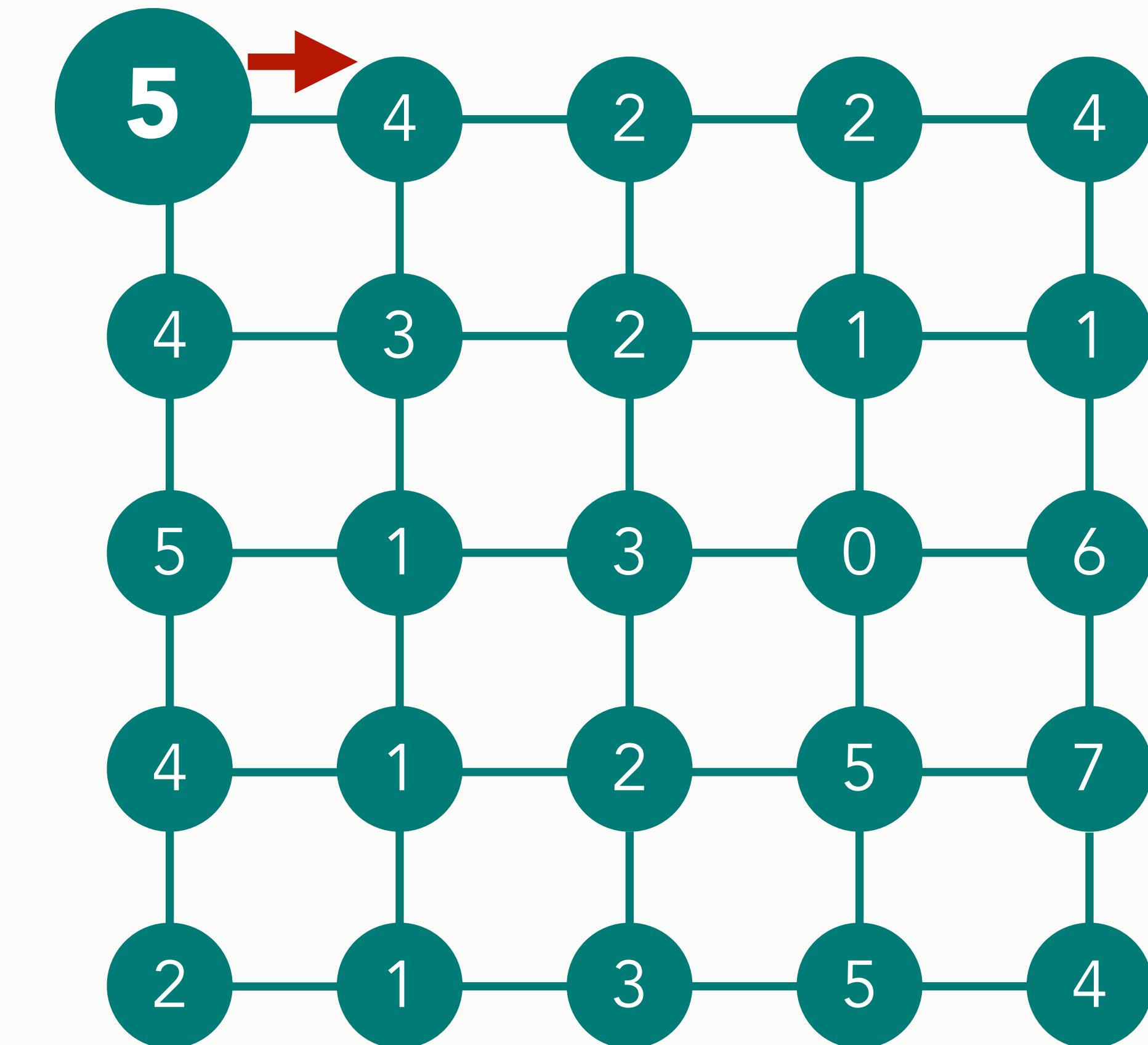
búsqueda local del primer mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local del  
primer mejor



# Métodos de búsqueda local básicos

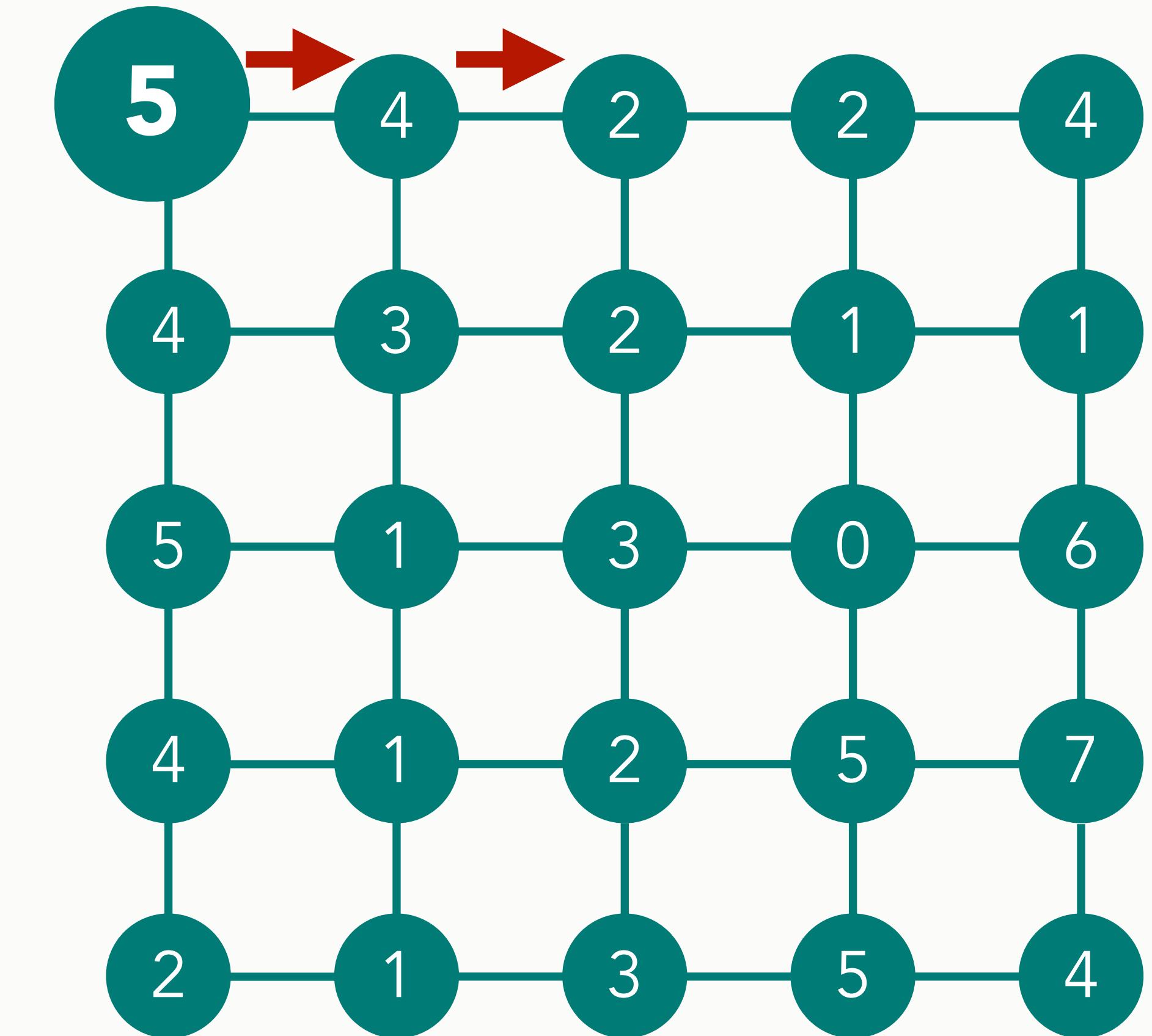
búsqueda local del primer mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local del  
primer mejor



# Métodos de búsqueda local básicos

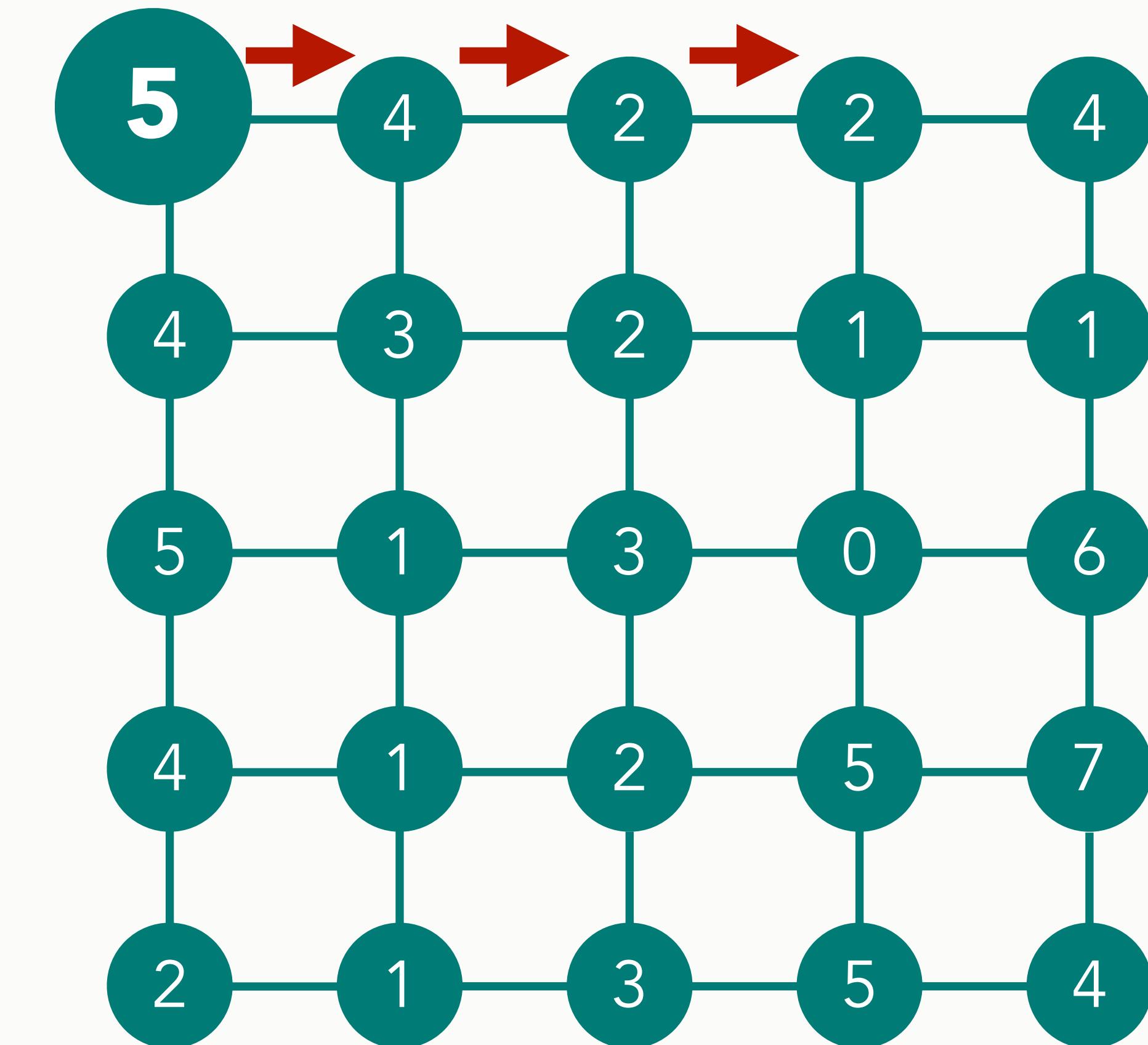
búsqueda local del primer mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

**Solución  
inicial**

**Trayectoria de la  
búsqueda local del  
primer mejor**



# Métodos de búsqueda local básicos

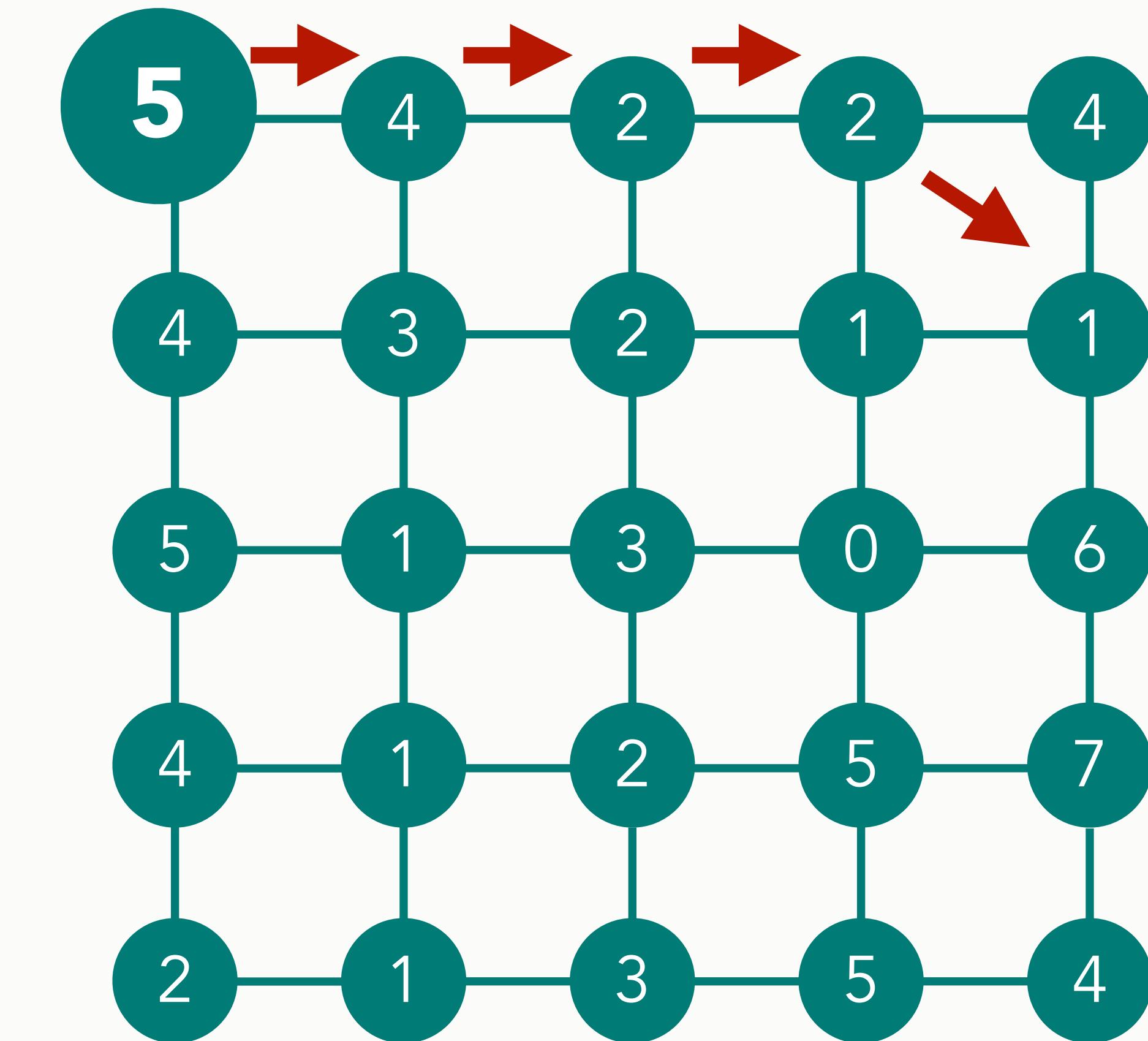
búsqueda local del primer mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

**Solución  
inicial**

**Trayectoria de la  
búsqueda local del  
primer mejor**



# Métodos de búsqueda local básicos

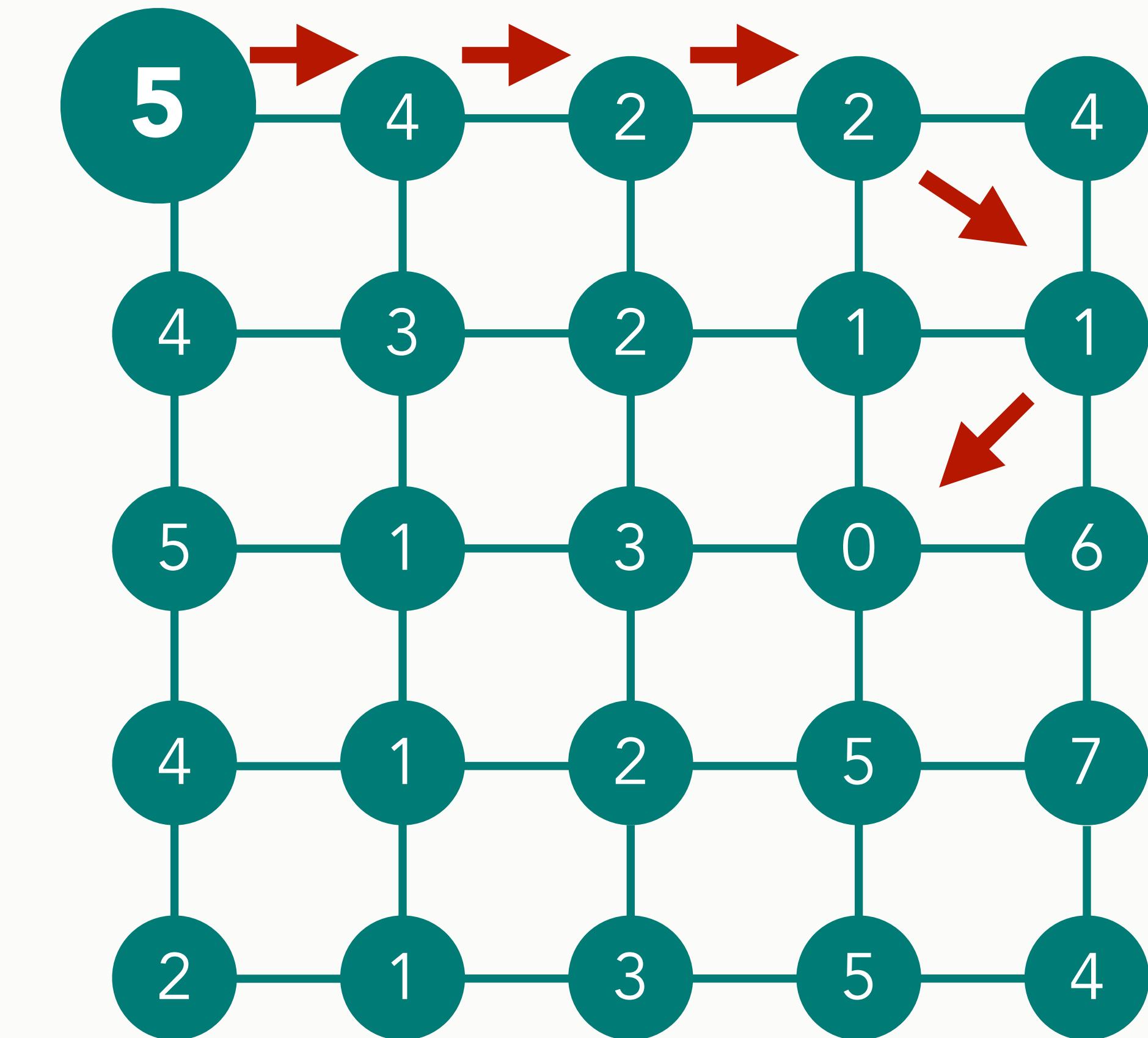
búsqueda local del primer mejor

Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de la  
búsqueda local del  
primer mejor



# Métodos de búsqueda local básicos

búsqueda local del primer mejor

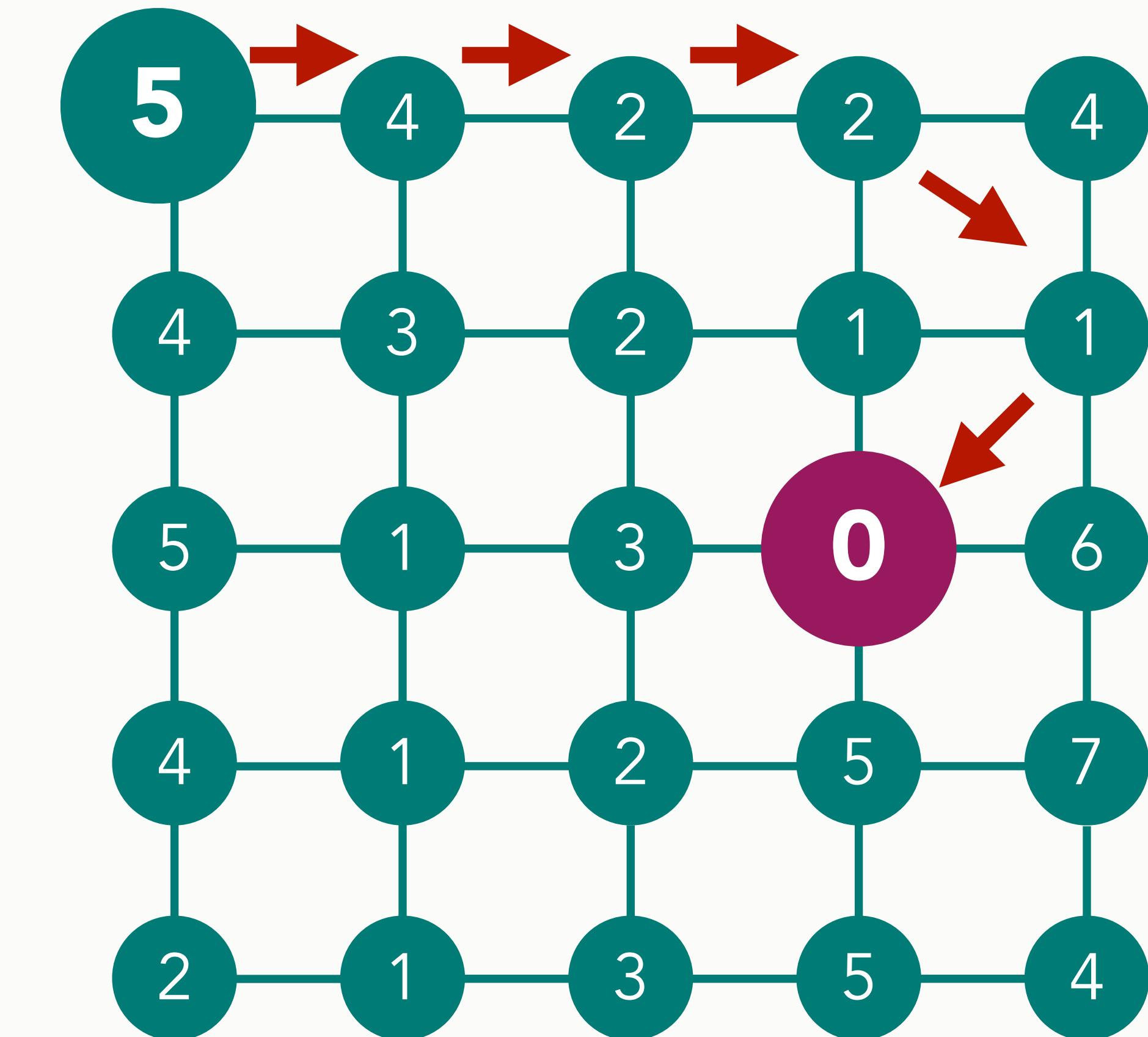
Problema es buscar el menor número partiendo de una casilla inicial

$$E(S) = \{S_i / S_i = (x_i \pm \{0,1\}, y_i \pm \{0,1\}) \wedge S_i \neq S\}$$

Solución  
inicial

Trayectoria de  
búsqueda local del  
primer mejor

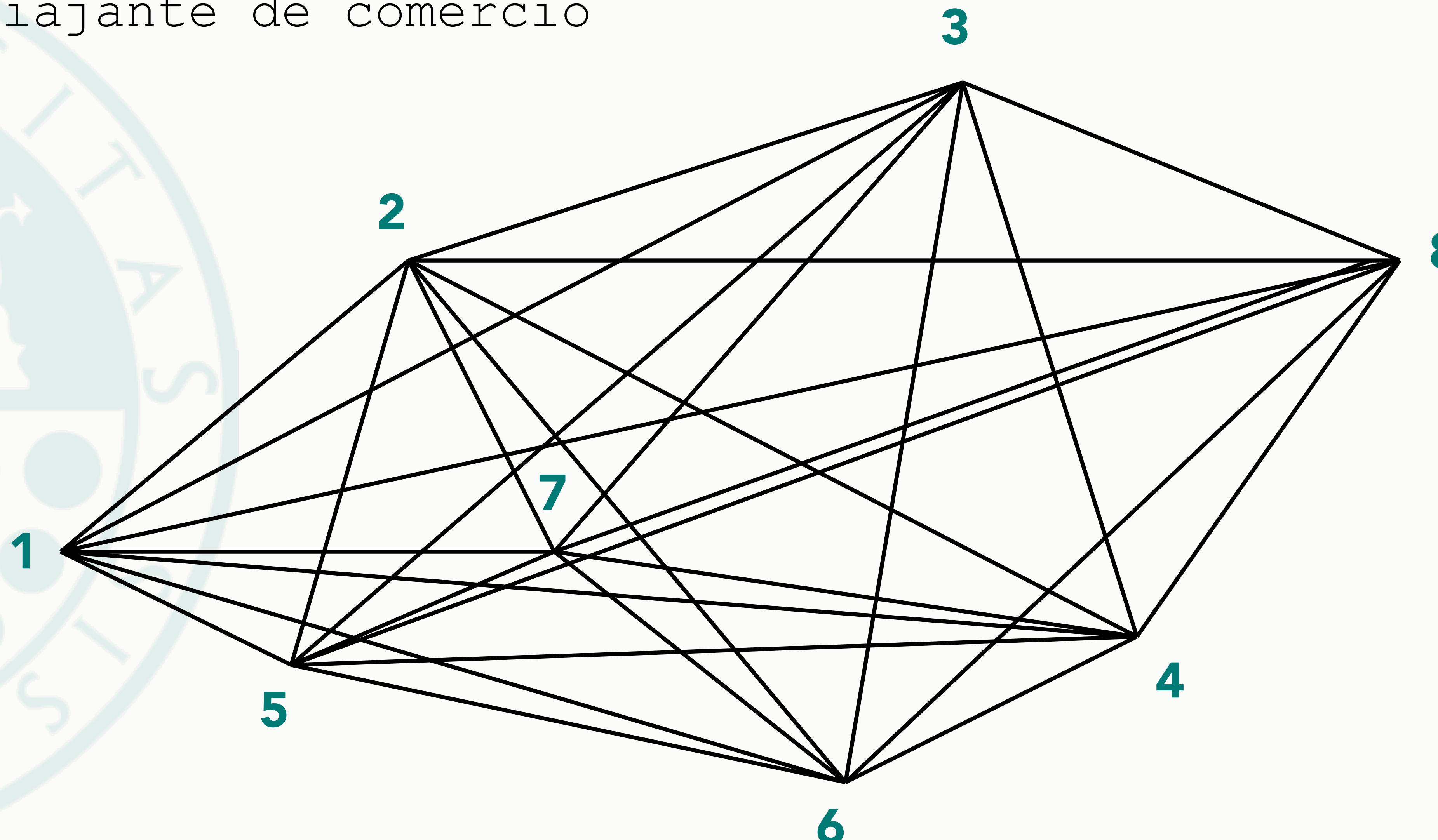
Solución  
devuelta



# Ejemplo de búsqueda local Viajante de comercio

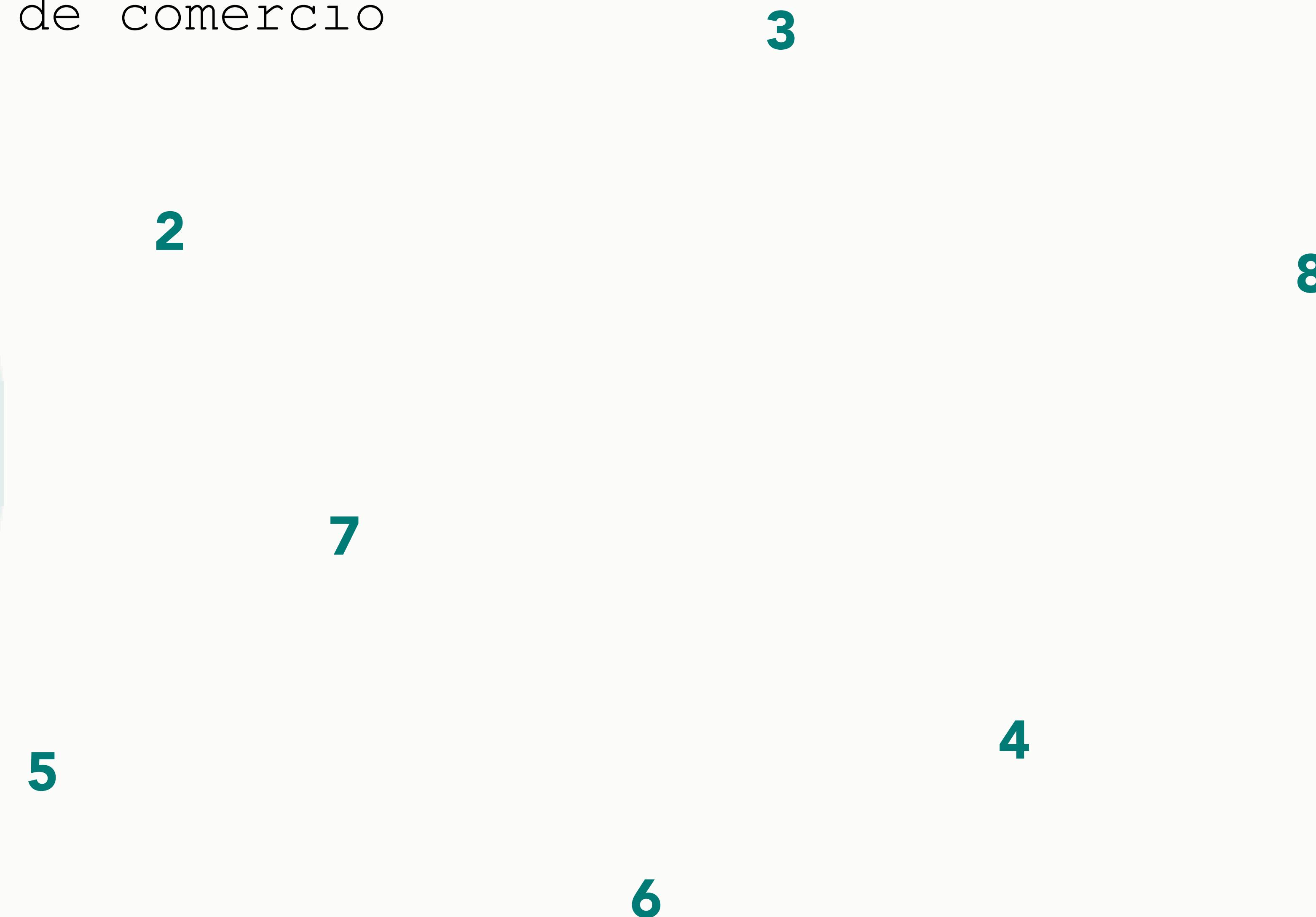
# Ejemplo de búsqueda local

el viajante de comercio



# Ejemplo de búsqueda local

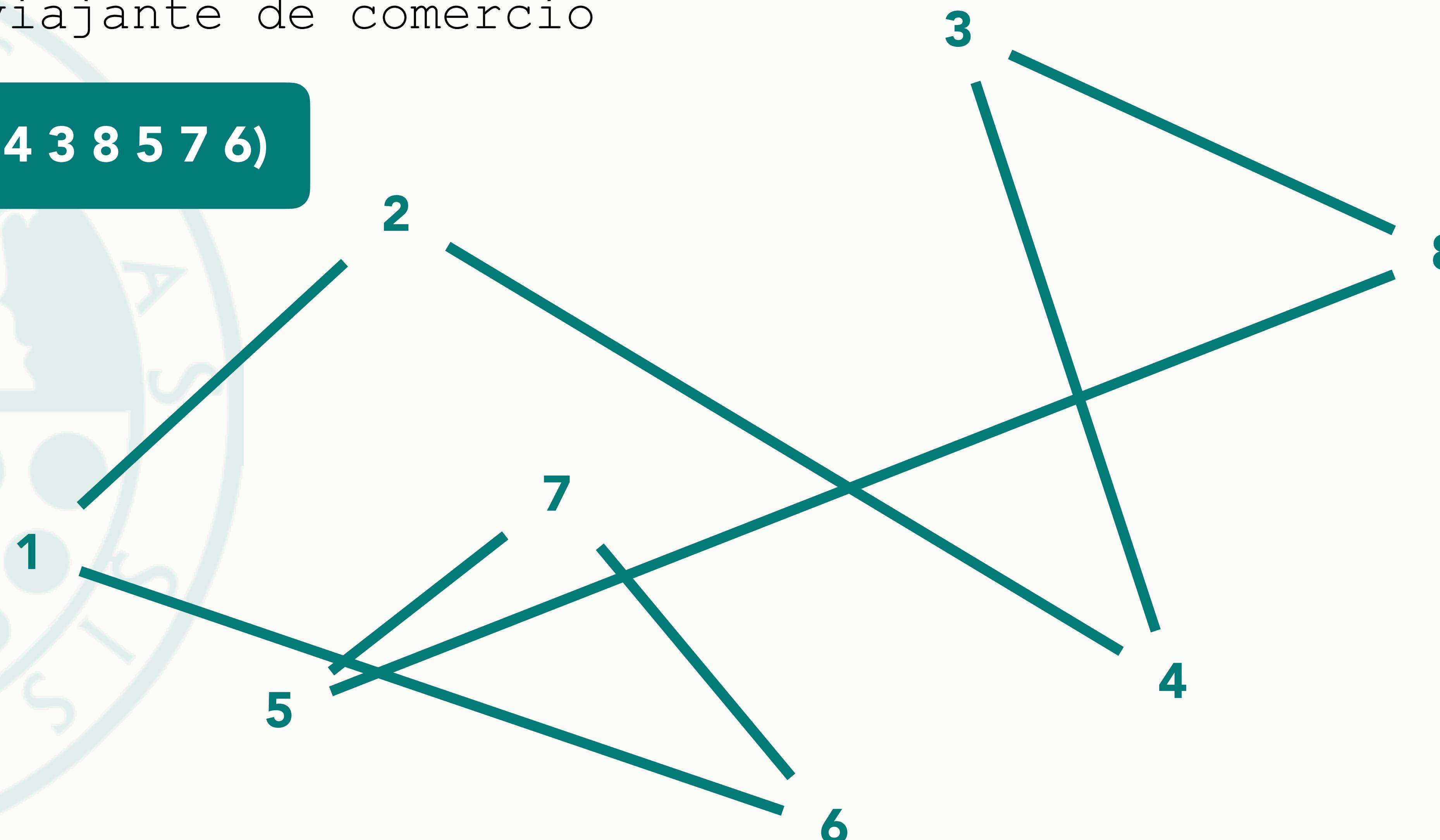
el viajante de comercio



# Ejemplo de búsqueda local

el viajante de comercio

(1 2 4 3 8 5 7 6)



# Ejemplo de búsqueda local

el viajante de comercio

**Esquema de representación mediante permutación {1, ..., n}**

**Función objetivo:**

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

Mecanismo de generación de la solución inicial mediante  
permutación aleatoria



# Ejemplo de búsqueda local

el viajante de comercio

## Entorno:

- **n-vecinos**: Se generan n soluciones vecinas a la solución actual
- **DLB**: Vector binario con bits activos
- ...



# Ejemplo de búsqueda local

el viajante de comercio

## Operador de generación de nuevas soluciones:

- **2-opt**: Se escogen dos posiciones e intercambian sus valores
- **Inversión simple**: Se escoge una lista y se invierte el orden

Mecanismo de selección del mejor o el primer mejor

Criterio de parada cuando el vecino no mejore a la solución actual

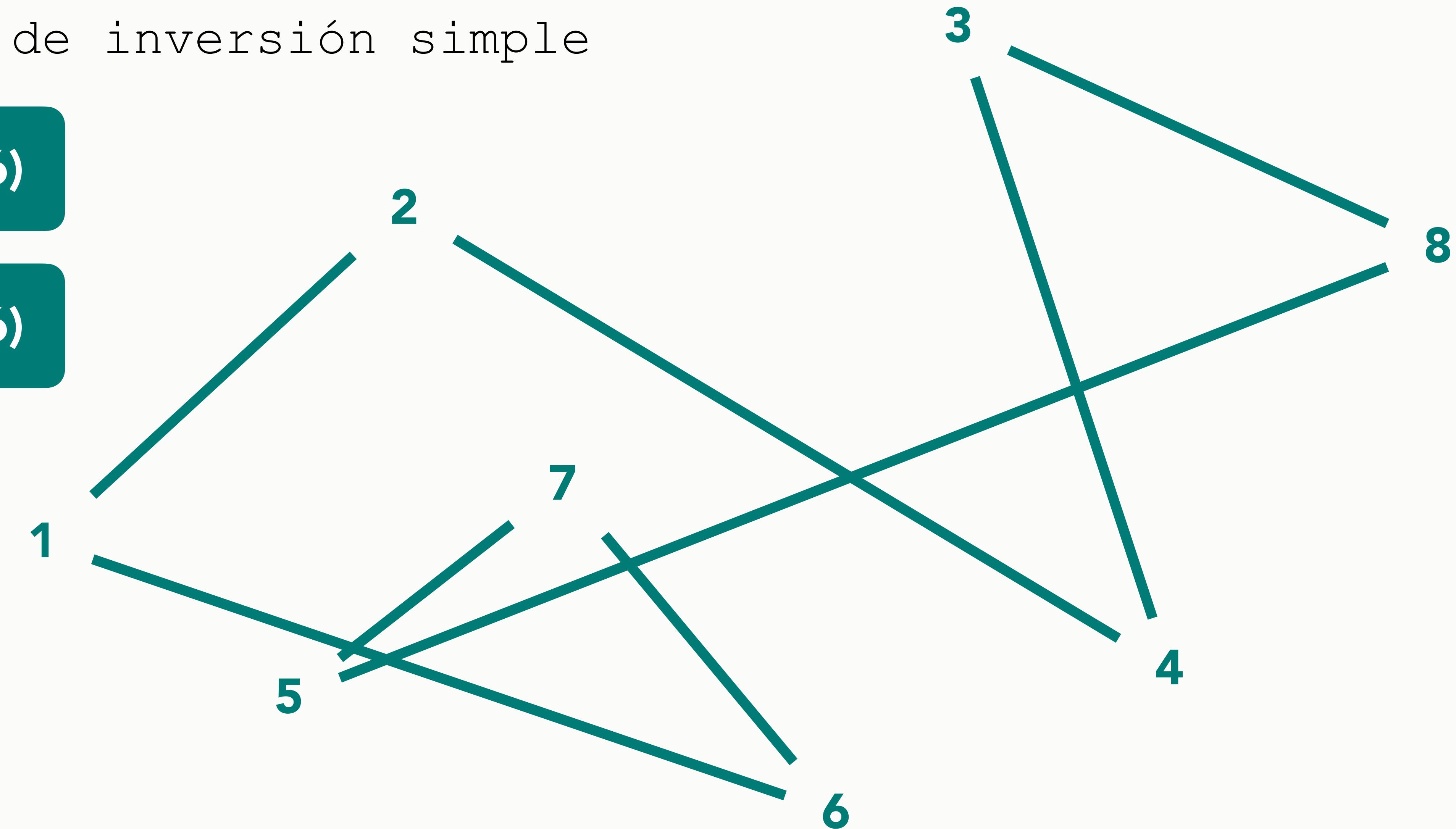


# Ejemplo de búsqueda local

el operador de inversión simple

(1 2 4 3 8 5 7 6)

(1 2 5 8 3 4 7 6)

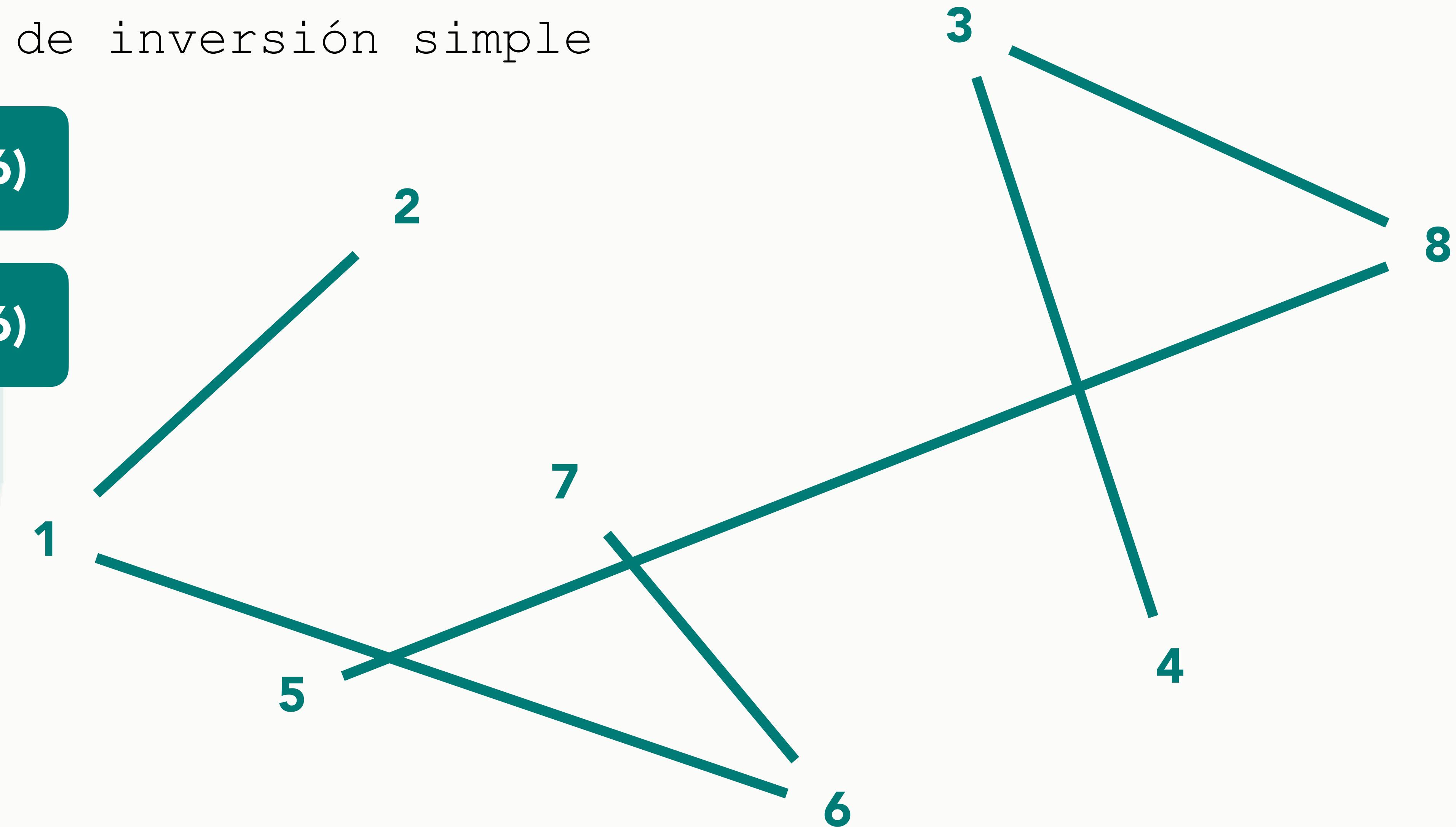


# Ejemplo de búsqueda local

el operador de inversión simple

(1 2 4 3 8 5 7 6)

(1 2 5 8 3 4 7 6)

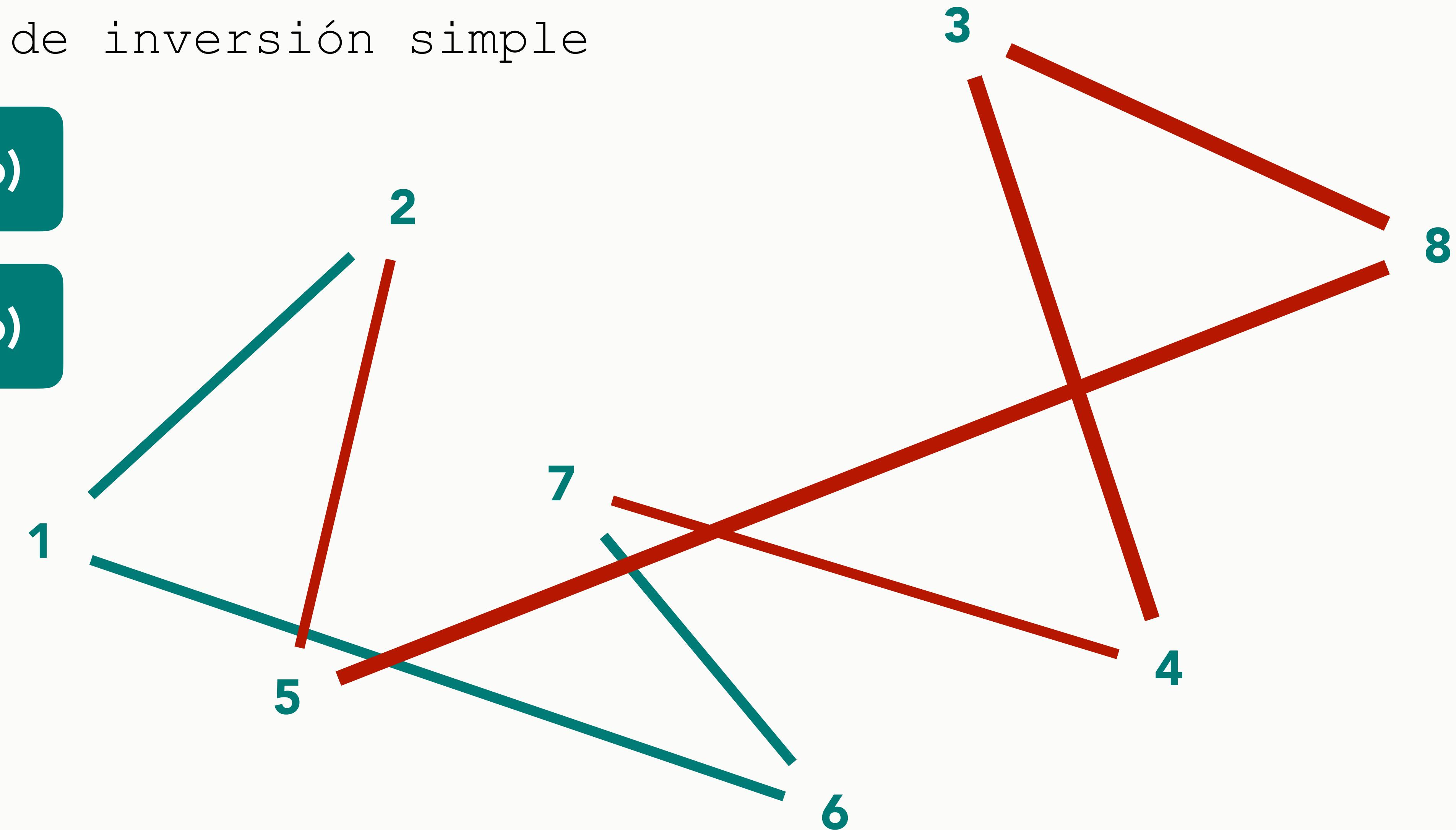


# Ejemplo de búsqueda local

el operador de inversión simple

(1 2 4 3 8 5 7 6)

(1 2 5 8 3 4 7 6)

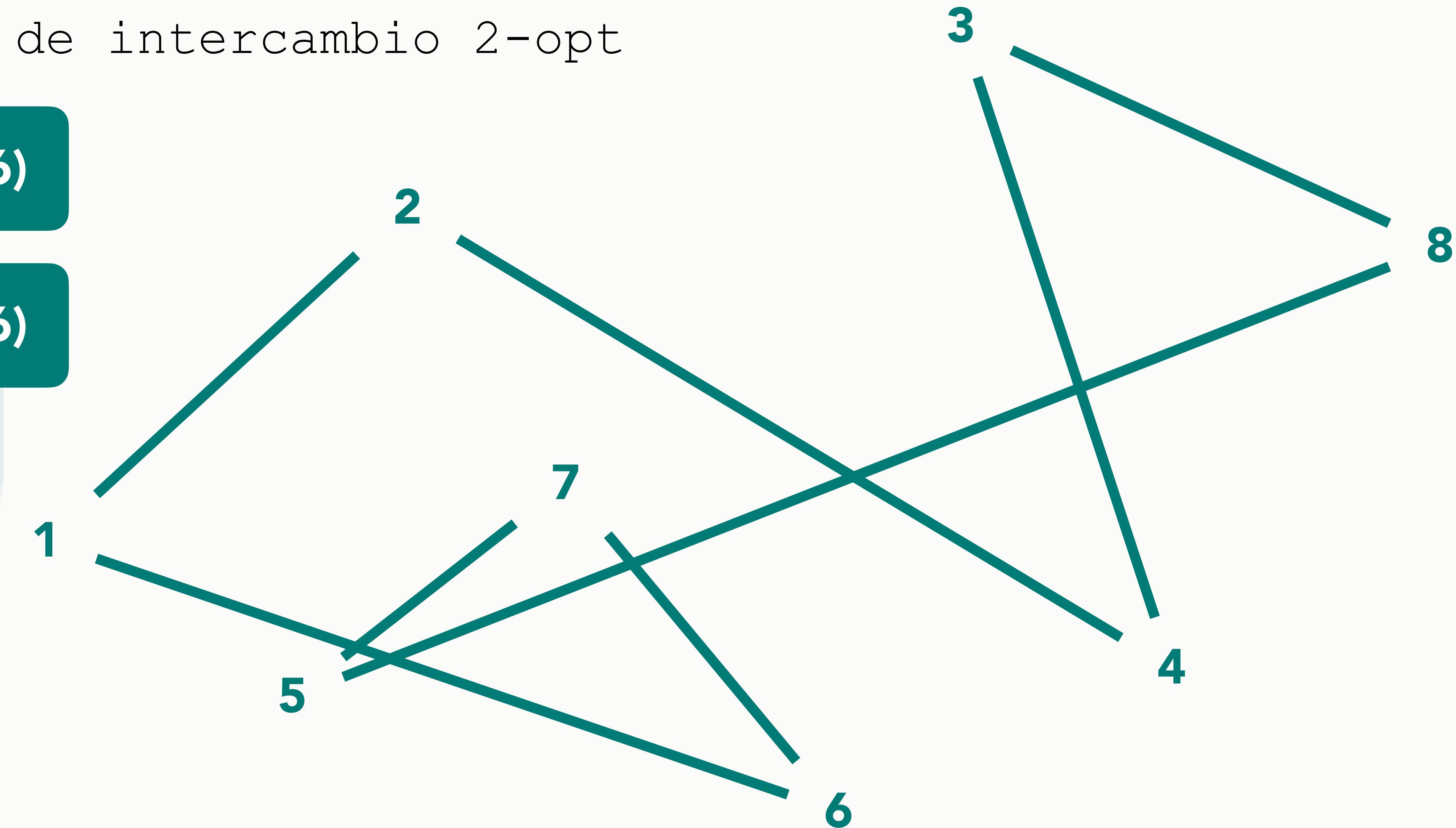


# Ejemplo de búsqueda local

el operador de intercambio 2-opt

(1 2 4 3 8 5 7 6)

(1 2 5 3 8 4 7 6)

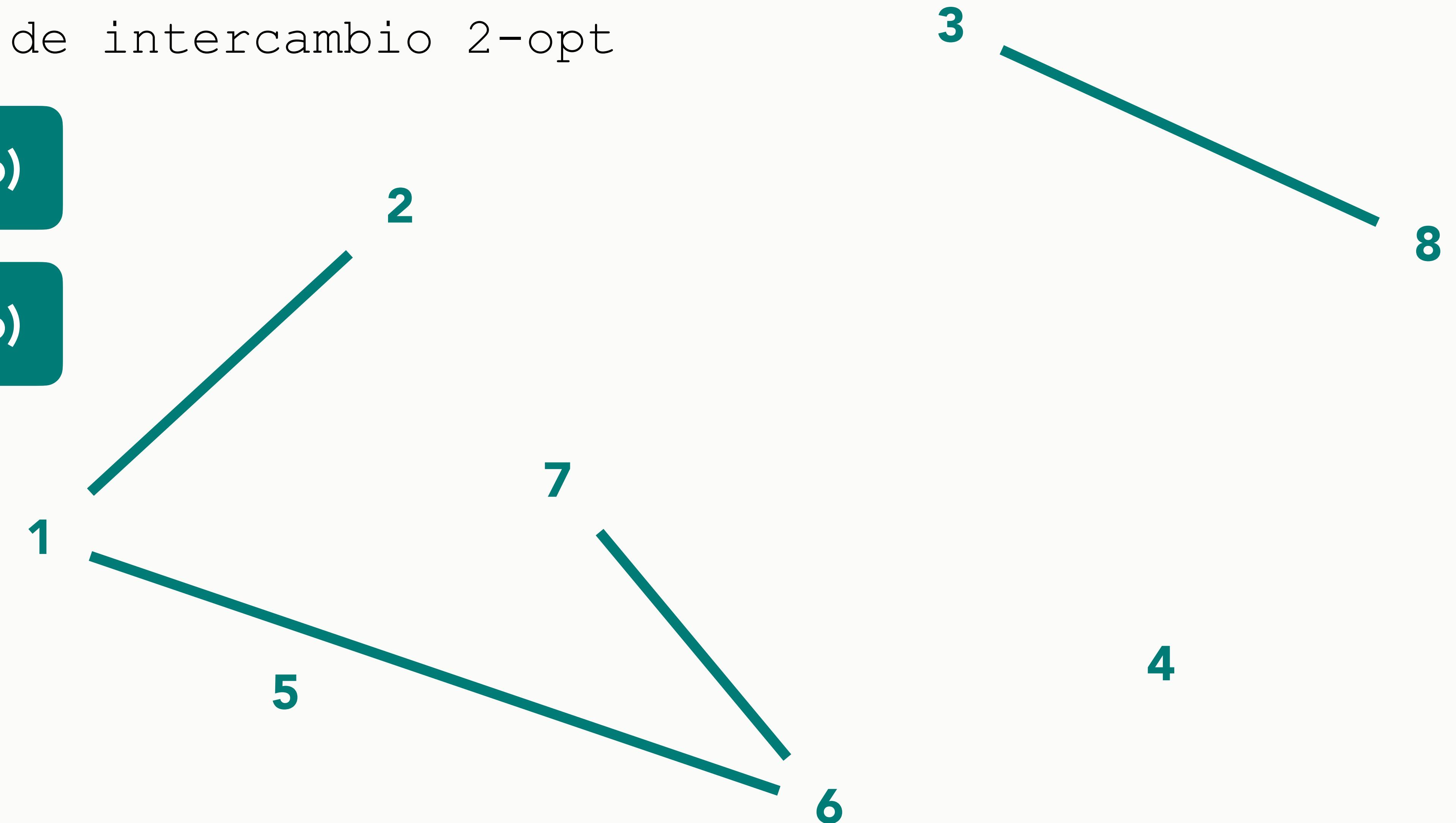


# Ejemplo de búsqueda local

el operador de intercambio 2-opt

(1 2 4 3 8 5 7 6)

(1 2 5 3 8 4 7 6)

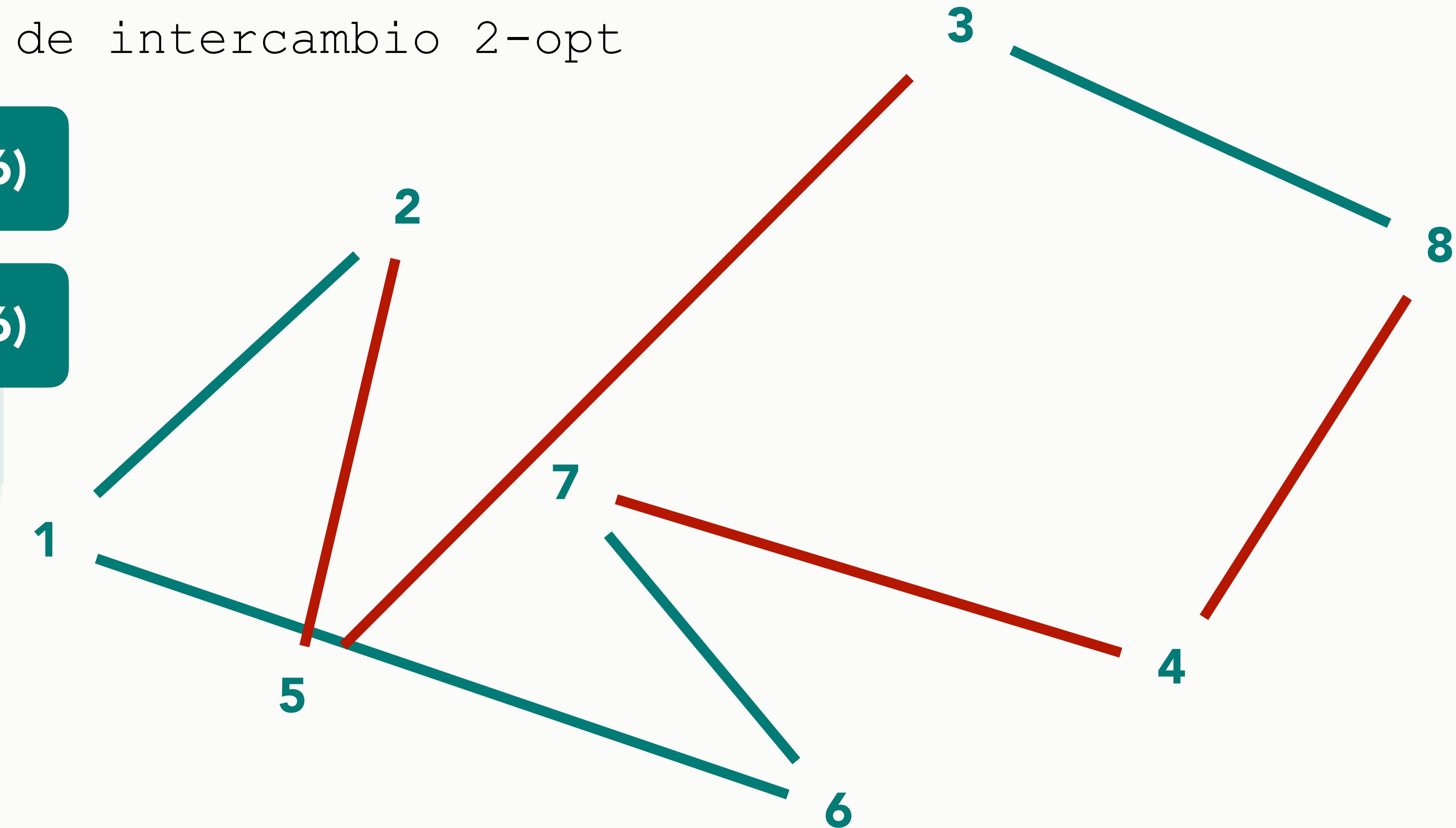


# Ejemplo de búsqueda local

el operador de intercambio 2-opt

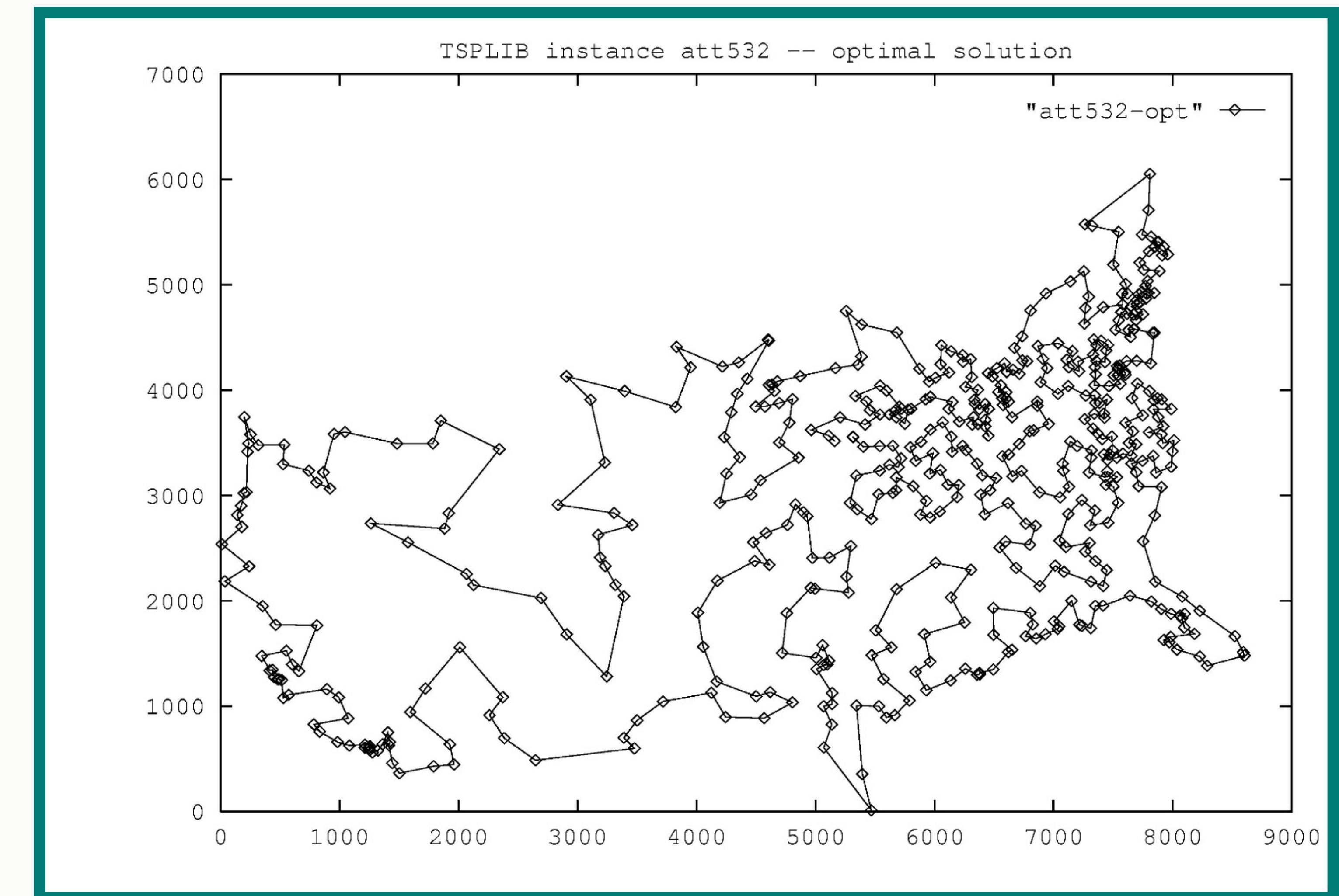
(1 2 4 3 8 5 7 6)

(1 2 5 3 8 4 7 6)



# Ejemplo de búsqueda local

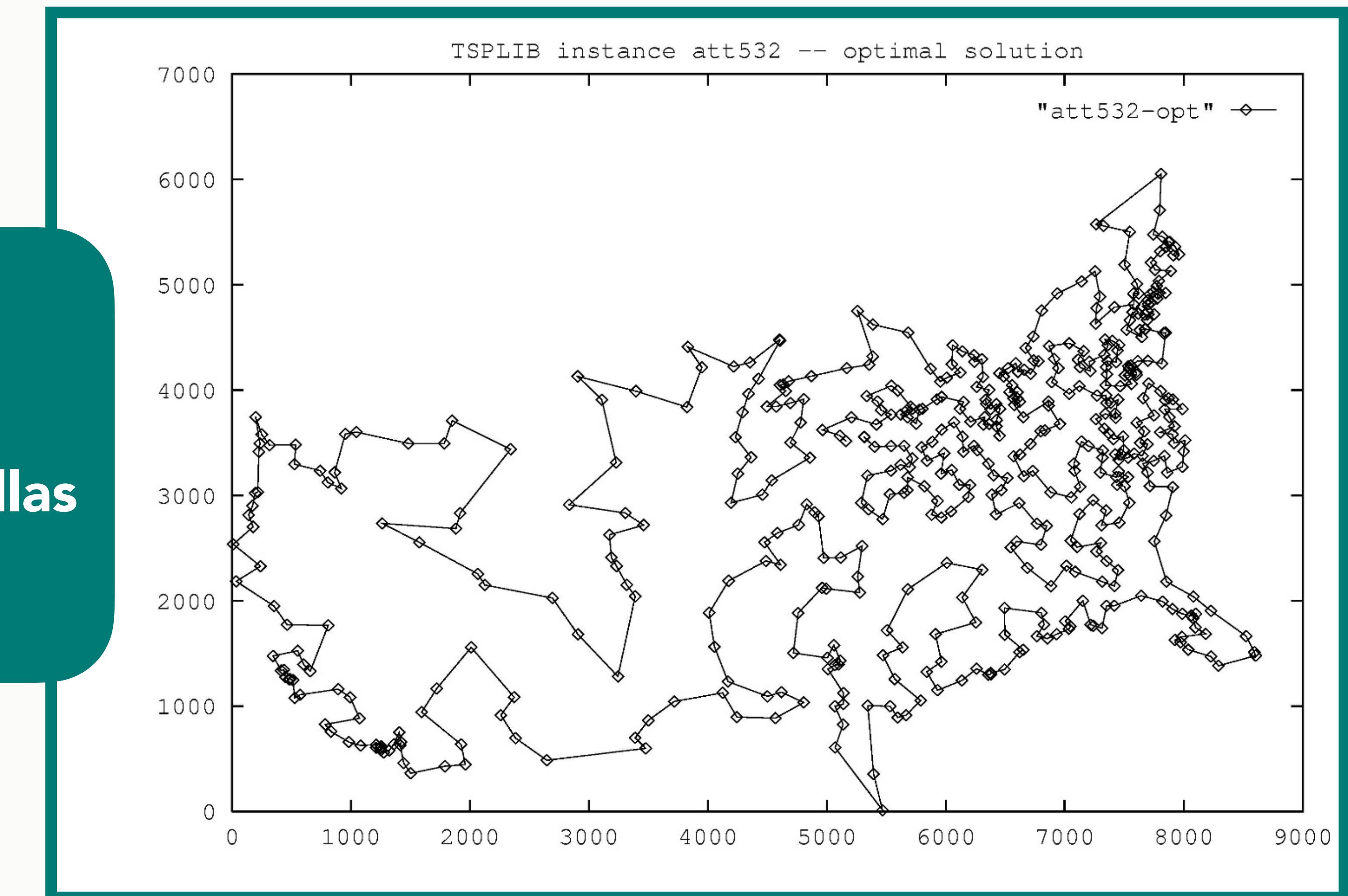
el viajante de comercio



# Ejemplo de búsqueda local

el viajante de comercio

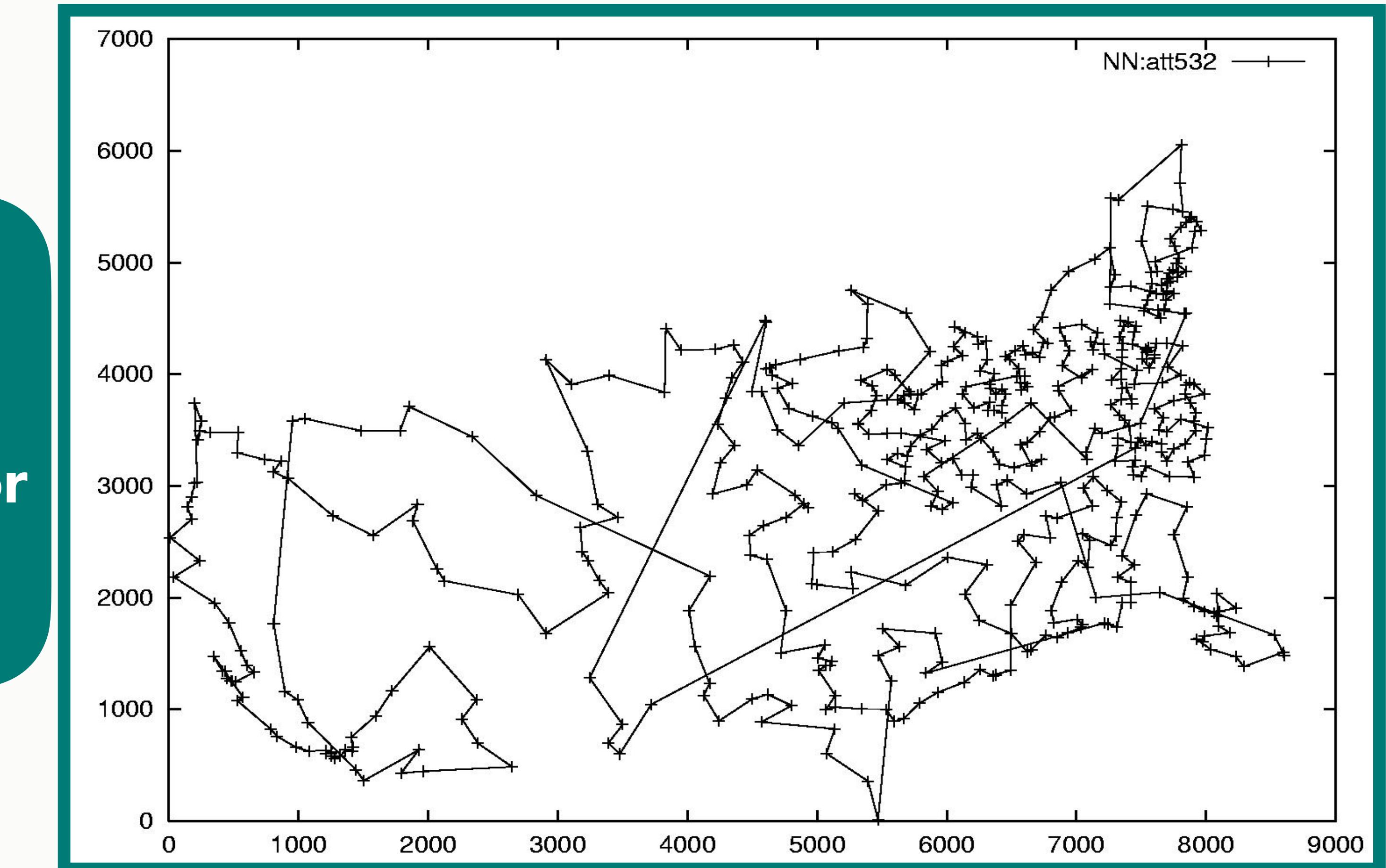
**532! soluciones posibles**  
**Solución óptima = 27686 millas**



# Ejemplo de búsqueda local

el viajante de comercio

**Solución obtenida mediante la  
búsqueda local del primer mejor**



# Preguntas

¿Cómo de diferente puede llegar a ser la solución vecina respecto de la solución actual según el operador de vecindario empleado?

¿Cuál es el tamaño del entorno generado por cada operador de vecino?

El efecto al aplicar un operador concreto, ¿es el mismo en todos los problemas?

Por ejemplo, ¿es lo mismo generar un vecino por intercambio de 2 posiciones en TSP que en otro problema?

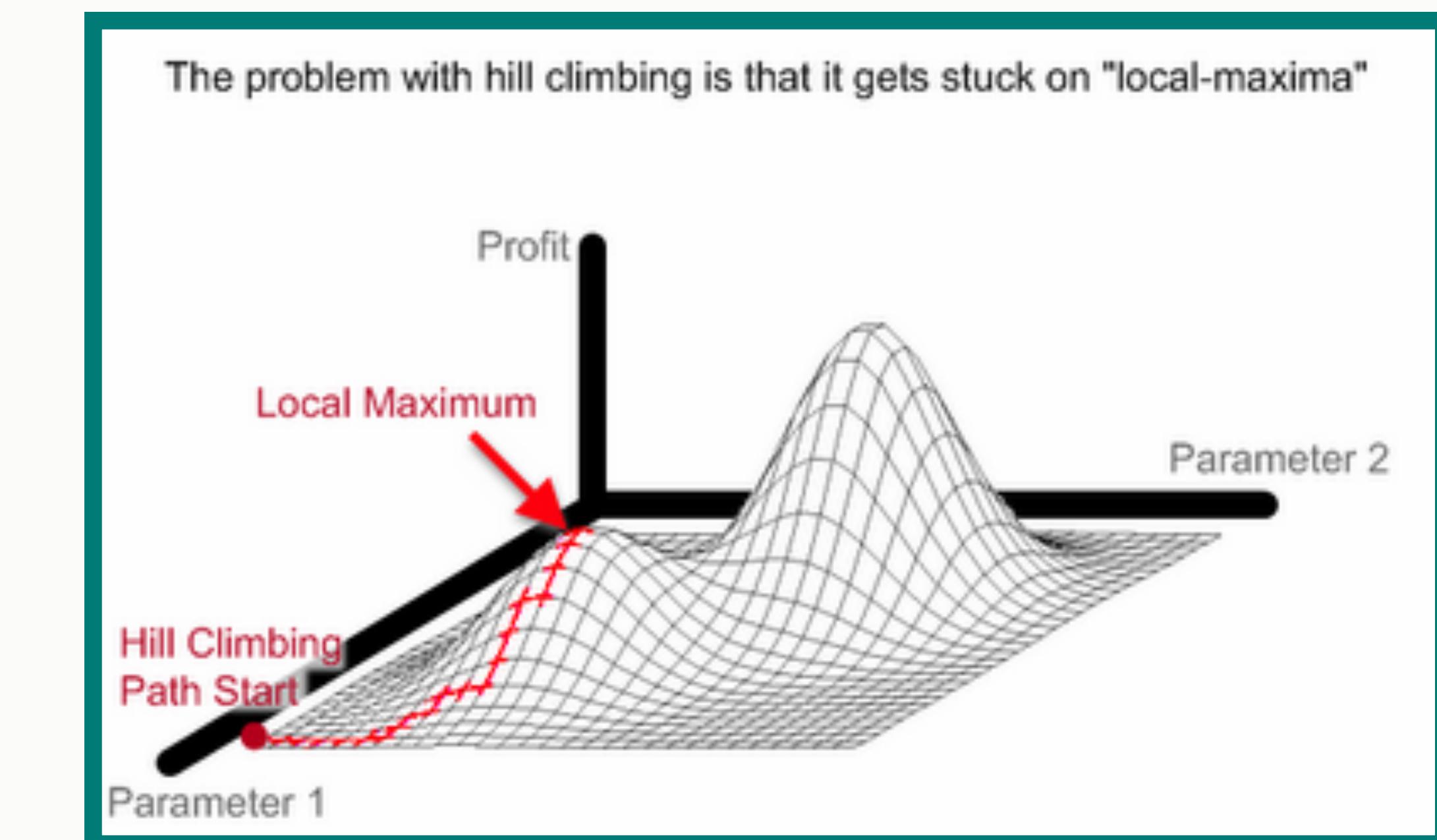
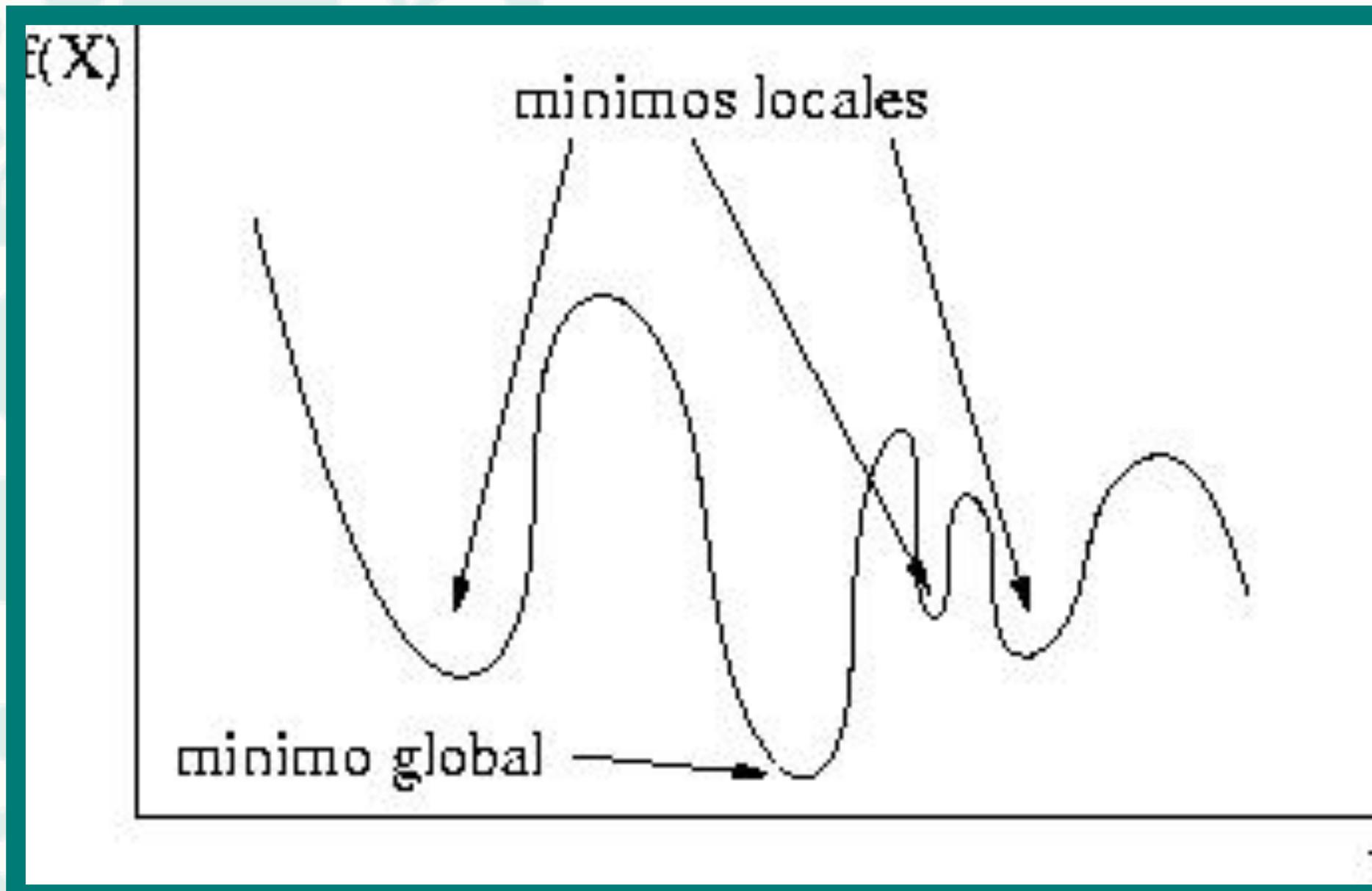
# 4

## Problemas de la búsqueda local

# Métodos de búsqueda local básicos

problemas de la búsqueda local

Suele caer en óptimos locales que a veces están bastante alejados del óptimo global del problema



# Métodos de búsqueda local básicos

problemas de la búsqueda local

Para salir de los óptimos locales se pueden emplear distintas soluciones:

1. Permitir movimientos de empeoramiento de la solución actual
2. Modificar la estructura de entornos
3. Volver a comenzar la búsqueda desde otra solución inicial

# Métodos de búsqueda local básicos

problemas de la búsqueda local

Para salir de los óptimos locales se pueden emplear distintas soluciones:

1. Permitir movimientos de empeoramiento de la solución actual
2. Modo de exploración
3. Volverse loco

**Enfriamiento simulado o la búsqueda tabú**

**Unidad 2**

# Métodos de búsqueda local básicos

problemas de la búsqueda local

Para salir de los óptimos locales se pueden emplear distintas

Búsqueda tabú o búsqueda en entornos variables  
Unidad 2

2. Modificar la estructura de entornos
3. Volver a comenzar la búsqueda desde otra solución inicial

# Métodos de búsqueda local básicos

problemas de la búsqueda local

Para salir de los óptimos locales se pueden emplear distintas soluciones:

1. R

act

2. N

Búsquedas multiarranque, ILS, VNS, ...

Unidad 2

3. Volver a comenzar la búsqueda desde otra solución inicial



# 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

