

# Búsquedas por entornos

---

Christopher Expósito Izquierdo, J. Marcos Moreno Vega  
cexposit@ull.es, jmmoreno@ull.es

Departamento de Ingeniería Informática y de Sistemas  
Universidad de La Laguna

# BÚSQUEDAS POR ENTORNOS

---

## Descripción

---

- Después de fijar una apropiada codificación para las soluciones y 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.
- El proceso de escoger una solución del entorno de la solución actual consta de dos fases: seleccionar la solución y decidir si se acepta o no.
- Otro elemento importante en tales búsquedas es el método por el cuál se determina la solución de partida para iniciar el recorrido.

## Búsqueda por entornos (NS)

---

**datos:**  $N(\cdot)$ , estructura de entorno  
Criterio de parada

**entrada:**  $X$ , solución de inicio

**salida:**  $X_{mej}$ , mejor solución encontrada

---

**Algorithm 1:**  $NS(X, X_{mej})$ 

---

**begin**

$X_{act} := X_{mej} := X;$

**repeat**

    Genera( $X_{vec} \in N(X_{act});$

**if** *Acepta*( $X_{vec}$ ) **then**

        |  $X_{act} := X_{vec}$

**end**

**if**  $f(X_{act}) < f(X_{mej})$  **then**

        |  $X_{mej} := X_{act}$

**end**

**until** *Criterio de parada*;

**return**  $X_{mej};$

**end**

---

## Elementos

---

- Estructura de entorno.
- Mecanismo de generación de soluciones iniciales.
- Muestreo en el entorno.
- Regla de parada.

## Mecanismo de generación de soluciones iniciales (i)

---

- **Deterministas:** el procedimiento de generación suministra siempre la misma solución.
- **Aleatorios:** el mecanismo de generación de soluciones depende completamente de sucesos aleatorios.
- **Mixtos:** se combinan los dos anteriores. Modificando ligeramente el primer procedimiento descrito se obtiene un mecanismo de generación mixto.

## Mecanismo de generación de soluciones iniciales (ii)

---

### $p$ -mediana discreta

- **Mecanismo determinístico:** Inicializar la solución  $X$  con el vértice con menor distancia promedio al resto de vértices. A continuación, tomar como mediana el vértice más alejado de  $X$ . Repetir este proceso hasta que  $X$  tenga  $p$  vértices.
- **Mecanismo aleatorio:** Formar  $X$  con  $p$  vértices seleccionados al azar.
- **Mecanismo mixto:** Inicializar la solución  $X$  con uno de los  $r$  (parámetro) vértices, seleccionado al azar, con menor distancia promedio al resto de vértices. A continuación, tomar como mediana el vértice más alejado de  $X$ . Repetir este proceso hasta que  $X$  tenga  $p$  vértices.

## Muestreo en el entorno

---

- **Muestreo greedy (Búsqueda del mejor):** se realiza una búsqueda exhaustiva en el entorno de la solución y se toma la mejor vecina.
- **Búsqueda del primer mejor:** la búsqueda en el entorno finaliza en cuanto se encuentra una solución mejor que la actual.
- **Muestreo aleatorio:** se escoge aleatoriamente una solución del entorno de la solución actual, o la mejor de una muestra seleccionada aleatoriamente del entorno.
- **Muestreo heurístico:** se toma aquella solución (o equivalentemente se realiza aquel movimiento) que, en base a una evaluación heurística, suministre una solución mejor que la actual.

## Muestreo en el entorno (ii)

---

### Ciclo mediano

- **Muestreo heurístico:** Al aplicar el movimiento de inserción, añadir al ciclo el vértice con mayor coste de asignación.

## Regla de parada (i)

---

- En cualquier procedimiento de búsqueda de soluciones para un problema dado, uno de los elementos más importantes es el criterio de parada empleado. La regla de parada es responsable, en gran medida, del grado de eficiencia y eficacia del procedimiento de solución.
- En ocasiones, el criterio de parada viene determinado por la búsqueda empleada. Así ocurre, por ejemplo, en una Búsqueda local.
- Otras veces se emplean criterios de parada como finalizar después de un tiempo de CPU fijado a priori, o después de un número fijo de evaluaciones de la función objetivo. Estos criterios suelen ser poco eficaces, dado que no emplean ninguna información sobre la evolución de la búsqueda.

## Regla de parada (ii)

---

- Se obtienen criterios de parada más eficaces al realizar un estudio del procedimiento, estudiar la función objetivo, estudiar la región factible, analizar la evolución de la búsqueda o evaluar las características de las soluciones obtenidas.
- En todo caso, cualquiera que sea la regla de parada empleada, esta debe asegurar un equilibrio entre eficiencia y eficacia.

## Regla de parada (iii)

---

- **Dependencia del problema:** si se conoce alguna propiedad de la región factible o de la función objetivo, esta debe usarse para obtener reglas de parada.
- Los y Lardinois [LOR82] estudian la distribución del valor objetivo de los óptimos locales de la función objetivo y obtienen reglas de parada empleando esta información.

## Regla de parada (iv)

---

- **Dependencia muestral:** cuando se ejecuta una heurística para resolver un problema, se obtiene información de varias variables: valor objetivo, distancia entre óptimos locales, tamaño de la región de atracción de los óptimos locales, iteraciones necesarias para obtener el óptimo global, etc. El análisis de estos valores puede suministrar reglas de parada apropiadas.
- Moreno-Pérez, Mladenović y Moreno-Vega [MOR95] aproximan la variable número de iteraciones necesarias para encontrar el óptimo global por medio de una distribución normal y emplean esta aproximación para obtener una regla de parada.

## Regla de parada (v)

---

- **Dependencia del método:** el estudio teórico de algunas heurísticas permite obtener reglas de parada que, al menos a nivel teórico, aseguran la convergencia al óptimo global del problema.
- Quizás el estudio más amplio en este sentido ha sido el realizado para el Recocido Simulado [AAR89].

## Regla de parada (vi)

---

- **Dependencia del costo y del recurso:** cuando se decide finalizar un procedimiento heurístico y se aporta como solución al problema la mejor solución encontrada, se incurre en dos pérdidas: una de finalización, que depende de la distancia entre el valor objetivo óptimo y aquel que suministra la heurística, y una de ejecución, que es función de la cantidad de recursos empleados. Estas pérdidas deben influir activamente en el criterio de parada. Obviamente, se pretende obtener una regla de parada que minimice ambas pérdidas.
- Boender y Rinnooy Kan [BOE87] realizan un amplio estudio de esta alternativa, y proponen y analizan diversas reglas de parada.

---

**Algorithm 2:** Búsqueda local( $X, X_{min}$ )

---

**datos** :  $N(\cdot)$ , estructura de entorno

**entrada:**  $X$ , solución de inicio

**salida** :  $X_{min}$ , mínimo obtenido

$X_{min} = X;$

**repeat**

    Genera( $X_{vec} \in N(X_{min}) / f(X_{vec}) < f(X_{min})$ );

$X := X_{vec};$

**until**  $f(X_{vec}) \geq f(X_{min}) \forall X_{vec} \in N(X_{min});$

**return**  $X_{min};$

---

---

**Algorithm 3:**  $p$ -medianaBL( $X, X_{min}$ )

---

$X_{min} \leftarrow X;$

**repeat**

$X' \leftarrow X_{min};$

    Mejor movimiento de intercambio

**for**  $1 < i \leq p, p < j \leq n$  **do**

$X_{ij} \leftarrow X' \setminus \{v_i\} \cup \{v_j\};$

**if**  $f(X_{ij}) < f(X_{min})$  **then**

$X_{min} \leftarrow X';$

**end**

**end**

**until**  $f(X') = f(X_{min});$

**return**  $X_{min};$

---

---

**Algorithm 4:** cicloMedianoBL( $X, X_{min}$ )

---

$r = 0;$

$X^r \leftarrow X;$

**repeat**

    Mejor movimiento de inserción o eliminación

$X' \leftarrow \arg \min \{f(X_i^r) : i = 2, \dots, n\};$

    Mejor movimiento de intercambio

$X'' \leftarrow \arg \min \{f(X_{ij}^r) : i = 2, \dots, p, j = p + 1, \dots, n\};$

$X^{r+1} \leftarrow \arg \min \{f(X'), f(X'')\};$

$r = r + 1;$

**until**  $f(X^{r-1}) = f(X^r);$

$X_{min} \leftarrow X^r;$

**return**  $X_{min};$

---

## Inconveniente

---

- Una Búsqueda local solo asegura optimalidad local.
- El óptimo local encontrado puede estar muy alejado del óptimo global.

## Alternativas

---

- Permitir de forma controlada movimientos o soluciones que supongan un empeoramiento (Simulated Annealing).
- Modificar la noción de vecindad empleada (Variable Neighborhood Search).
- Aplicar búsquedas locales desde diferentes soluciones de inicio (Búsquedas multiarranque).

- [AAR89] **Simulated Annealing and Boltzmann Machines.** Aarts, E, Korst, J. *John Wiley and Sons* (1989)
- [BOE87] **Bayesian Stopping Rules for Multistart Global Optimization Methods.** Boender, C.G.E., Rinnooy Kan, A.H.G. *Mathematical Programming* 37 (1987) pp. 59-80
- [LOR82] **Combinatorial Programming, Statistical Optimization and the Optimal Transportation Network Problem.** Los, M., Lardinois, C. *Transportation Research*, 2 (1982) pp. 89-124
- [MOR95] **An statistical analysis of strategies for multistart heuristic searches for p-facility location-allocation problems.** Moreno-Pérez, J.A., Mladenović, N., Moreno-Vega, J. M. *8th Meeting of the Euro Working Group on Locational Analysis*, Lambrecht, Alemania (1995)



Este obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.