

**Materia:**

**DISEÑO ELECTRÓNICO BASADO EN  
SISTEMAS EMBEBIDOS**

**Alumno:**

Posadas Pérez Isaac Sayeg

Paniagua Rico Juan Julian

García Azzúa Jorge Roberto

**Grado y grupo:**

8°G

**Profesor:**

Garcia Ruiz Alejandro Humberto

**Unidad 3 - Tarea 11:**

Búsqueda tabú

# Búsqueda Tabú en Optimización para Sistemas Embebidos

## Introducción

La **búsqueda tabú** es una técnica metaheurística de optimización diseñada para escapar de los óptimos locales que suelen atrapar a métodos como la búsqueda local. Introduce una memoria adaptativa que guía la exploración del espacio de soluciones, evitando volver a soluciones previamente visitadas mediante el uso de una lista tabú.

En el contexto de los sistemas embebidos, donde las soluciones deben ser eficientes y rápidas, la búsqueda tabú se presenta como una alternativa poderosa para problemas de planificación, asignación de recursos, diseño de hardware y optimización de consumo energético.

## Desarrollo

### 1. ¿Qué es la búsqueda tabú?

Es un método de búsqueda local avanzada que incorpora memoria a corto y largo plazo para registrar movimientos o soluciones recientes y prohibir (hacer "tabú") ciertos cambios, con el fin de explorar regiones nuevas del espacio de búsqueda.

Su objetivo es evitar ciclos y salir de óptimos locales, mejorando la calidad global de la solución.

### 2. Elementos clave de la búsqueda tabú

#### a) Solución actual y vecindario

La búsqueda parte de una solución inicial  $xxx$ , y en cada iteración examina su vecindario  $N(x)N(x)N(x)$ , es decir, soluciones alcanzables con una pequeña modificación.

#### b) Lista tabú

Es una estructura de memoria que guarda los últimos movimientos (o atributos de soluciones) realizados, para evitar deshacerlos en las siguientes iteraciones. Por ejemplo:

$$\text{Tabú} = \{(x_i \rightarrow x_j), (x_k \rightarrow x_l), \dots\}$$

Tiene una longitud limitada TTT y opera como una cola FIFO (el más antiguo se elimina).

#### c) Criterio de aspiración

Permite ignorar una restricción tabú si la nueva solución es significativamente mejor que la actual, evitando que la lista tabú impida mejoras.

#### d) Función objetivo

$$f(x) = \text{costo total, consumo, tiempo, etc.}$$

La calidad de cada solución se mide con una función objetivo que depende del problema.

### 3. Algoritmo básico de búsqueda tabú

1.  $x \leftarrow \text{GenerarSoluciónInicial}()$
2.  $x^* \leftarrow x$
3.  $\text{ListaTabú} \leftarrow \emptyset$
4. Mientras no se cumpla la condición de parada:
  - a.  $N(x) \leftarrow \text{generar vecindario de } x$
  - b.  $x' \leftarrow \text{mejor vecino de } N(x) \text{ no tabú o permitido por criterio de aspiración}$
  - c.  $x \leftarrow x'$
  - d. actualizar  $\text{ListaTabú}$  con movimiento realizado
  - e. si  $f(x) < f(x^*)$ :  $x^* \leftarrow x$
5. devolver  $x^*$

### 4. Aplicaciones en sistemas embebidos

#### a) Asignación de tareas en sistemas tiempo real

Optimiza el orden de ejecución para reducir tiempos de espera y cumplir deadlines.

#### b) Optimización de rutas en redes de sensores

Evita congestión o consumo excesivo de energía, mejorando la duración de la red.

#### c) Diseño de hardware configurable (FPGA)

Busca combinaciones eficientes de módulos bajo restricciones de área, energía y velocidad.

#### d) Gestión dinámica de energía

Selecciona configuraciones del sistema (frecuencia, voltaje) para minimizar consumo sin afectar rendimiento.

## 5. Ventajas y desventajas

Ventajas	Desventajas
Escapa de óptimos locales	Mayor complejidad de implementación
Usa memoria adaptativa	Requiere más espacio que búsqueda local
Mejora la exploración	Parámetros sensibles (tamaño lista tabú)
Puede combinarse con otras	

## 6. Ejemplo numérico simple (abstracto)

Supón que tienes una red de sensores y buscas minimizar la energía total consumida. La función objetivo puede ser:

$$f(x) = \sum_{i=1}^n E_i(x)$$

Donde  $E_i(x)$  es el consumo energético del nodo  $i$  bajo la configuración  $x$ .

- En cada iteración, se modifica ligeramente la asignación de tareas entre nodos.
- Se guarda en la lista tabú el movimiento: "mover tarea T de nodo A a B".
- Si esa opción se vuelve a considerar, se prohíbe por algunas iteraciones.

Si una nueva configuración logra un consumo energético muy bajo (mejor que el actual mínimo), se permite incluso si el movimiento está en la lista tabú (criterio de aspiración).

## Conclusión

La búsqueda tabú es una estrategia poderosa para resolver problemas de optimización difíciles, donde otros métodos quedan atrapados en óptimos locales. Su capacidad para usar memoria, evitar repeticiones y guiar la búsqueda hacia regiones inexploradas es especialmente útil en entornos embebidos con restricciones críticas.

Aunque implica un mayor esfuerzo de implementación y ajuste de parámetros, los beneficios en calidad de la solución y capacidad de adaptación la hacen ideal para sistemas embebidos inteligentes.

## Bibliografía

- ❖ Glover, F. (1989). *Tabu Search—Part I*. ORSA Journal on Computing.
- ❖ Glover, F., & Laguna, M. (1997). *Tabu Search*. Kluwer Academic.
- ❖ Talbi, E. G. (2009). *Metaheuristics: From Design to Implementation*. Wiley.
- ❖ Marwedel, P. (2011). *Embedded System Design*. Springer.
- ❖ Givargis, T., & Vahid, F. (2002). *Embedded System Design*. Wiley.