

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 9:

Búsqueda local

Búsqueda Local en Optimización para Sistemas Embebidos

Introducción

La búsqueda local es una técnica de optimización utilizada para resolver problemas complejos mediante la exploración de soluciones vecinas a una solución actual. Esta técnica es especialmente útil en el diseño y operación de sistemas embebidos, donde se requiere eficiencia en tiempo y recursos para tomar decisiones óptimas.

A diferencia de otros enfoques que buscan explorar todo el espacio de soluciones, la búsqueda local se enfoca en mejorar iterativamente una única solución, lo que la hace ideal para dispositivos con recursos limitados. Aunque puede quedar atrapada en óptimos locales, su simplicidad, rapidez y adaptabilidad la convierten en una herramienta fundamental en este campo.

Desarrollo

1. ¿Qué es la búsqueda local?

La búsqueda local es un método de optimización que parte de una solución inicial y explora su vecindario (conjunto de soluciones similares) para encontrar una mejor. Este proceso se repite hasta que no se encuentren mejoras o se cumpla un criterio de parada.

2. Estructura básica de una búsqueda local

1. **Generación de solución inicial**
Puede ser aleatoria o basada en alguna heurística.
2. **Evaluación del vecindario**
Se examinan soluciones cercanas a la actual.
3. **Selección de mejor vecino**
Se elige la mejor solución vecina que mejora la actual.
4. **Actualización de la solución actual**
Se reemplaza la solución si el vecino es mejor.
5. **Repetición o finalización**
Se detiene si no hay mejoras o se alcanza una condición de salida.

3. Tipos de búsqueda local

a) Primera mejora

Se acepta la primera solución mejor que la actual encontrada en el vecindario.

b) Mejor mejora

Se evalúan todas las soluciones vecinas y se elige la mejor de todas.

c) Búsqueda local aleatorizada

Incorpora decisiones probabilísticas para seleccionar vecinos, favoreciendo la exploración del espacio.

d) Búsqueda local iterada

Reinicia el proceso desde diferentes puntos para evitar quedar atrapado en óptimos locales.

4. Aplicaciones en sistemas embebidos

La búsqueda local se aplica con éxito en tareas como:

- **Asignación de tareas y planificación:**
Optimiza el orden y distribución de tareas en sistemas de tiempo real.
- **Ajuste de parámetros de control:**
Encuentra configuraciones óptimas de controladores PID embebidos.
- **Gestión de energía:**
Ajusta dinámicamente el consumo de energía frente al rendimiento.
- **Optimización de rutas de sensores móviles:**
Encuentra trayectorias eficientes con baja complejidad de cálculo.

5. Ventajas y desventajas

Ventajas:

- Bajo consumo de memoria y CPU.
- Fácil de implementar.
- Rápida convergencia a soluciones aceptables.

Desventajas:

- Puede estancarse en óptimos locales.
- La calidad de la solución depende de la inicial.
- Requiere diseño cuidadoso del vecindario y de la función objetivo.

Ejemplos prácticos

Ejemplo 1:

Sistema embebido para control de tráfico vehicular

Una búsqueda local ajusta en tiempo real la sincronización de semáforos para reducir congestión, evaluando pequeños cambios en tiempos por intersección.

Ejemplo 2:

Optimización de frecuencia dinámica en un microcontrolador

Se evalúan frecuencias vecinas a la actual para encontrar el mejor compromiso entre rendimiento y consumo energético.

Ejemplo 3:

Distribución de tareas en una red de sensores

Cada nodo busca ajustes mínimos en su tiempo de muestreo para reducir colisiones y prolongar la vida útil.

7. Mejoras comunes

- **Recocido simulado:**
Acepta, con cierta probabilidad, soluciones peores para escapar de óptimos locales.
- **Búsqueda tabú:**
Impide volver a soluciones recientes ya visitadas.
- **Hibridación:**
Se integra como parte de algoritmos evolutivos o metaheurísticos más complejos.

Conclusión

La búsqueda local representa una estrategia eficiente y práctica para resolver problemas de optimización en sistemas embebidos, especialmente cuando los recursos computacionales son escasos y se necesita una respuesta rápida. Aunque tiene limitaciones, como la posibilidad de quedar atrapada en óptimos locales, su rendimiento puede mejorarse mediante técnicas complementarias como recocido simulado o búsqueda tabú.

Su simplicidad y adaptabilidad la convierten en una herramienta muy valiosa en el diseño de soluciones embebidas inteligentes, donde el compromiso entre calidad, velocidad y consumo es clave.

Bibliografía

- ❖ Aarts, E., & Lenstra, J. K. (1997). *Local Search in Combinatorial Optimization*. Wiley.
- ❖ Marwedel, P. (2011). *Embedded System Design*. Springer.
- ❖ Givargis, T., & Vahid, F. (2002). *Embedded System Design*. Wiley.
- ❖ Talbi, E. G. (2009). *Metaheuristics: From Design to Implementation*. Wiley.
- ❖ Kirkpatrick, S., et al. (1983). *Optimization by Simulated Annealing*.
- ❖ Cormen, T. et al. (2009). *Introduction to Algorithms*. MIT Press.