

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 5:

Heurísticas

Heurística

Introducción

La heurística es una estrategia de resolución de problemas que busca soluciones satisfactorias dentro de un tiempo razonable, especialmente cuando una solución óptima exacta es difícil o imposible de obtener debido a la complejidad del problema o a las limitaciones del sistema. En sistemas embebidos, donde los recursos computacionales, de energía y de memoria son limitados, las heurísticas resultan particularmente útiles.

A diferencia de los algoritmos exactos, las heurísticas no garantizan encontrar la mejor solución, pero permiten alcanzar resultados suficientemente buenos en tiempos mucho menores. Por ello, se aplican en numerosas áreas del diseño embebido, como la planificación de tareas, la asignación de recursos, la compresión de datos, el enrutamiento en redes de sensores y la optimización de parámetros de control.

Desarrollo

1. Concepto de heurística

Las heurísticas son reglas, procedimientos o métodos prácticos que ayudan a resolver problemas complejos con base en el conocimiento empírico o la intuición. Se centran en obtener una “buena” solución dentro de límites razonables de tiempo y recursos, en lugar de buscar la perfección.

En contextos embebidos, esto puede significar que una solución heurística logra mantener el sistema funcionando dentro de especificaciones sin agotar los recursos disponibles ni violar restricciones de tiempo real.

2. Ventajas de las heurísticas en sistemas embebidos

- **Bajo uso de recursos:** Pueden ejecutarse en microcontroladores con capacidades limitadas.
- **Alta velocidad de cómputo:** Ofrecen respuestas rápidas para problemas que deben resolverse en tiempo real.
- **Flexibilidad:** Pueden adaptarse fácilmente a distintos escenarios y restricciones.
- **Robustez:** Pueden funcionar razonablemente bien incluso con datos incompletos o inciertos.

3. Aplicaciones comunes en sistemas embebidos

- **Planificación de tareas:** En sistemas de tiempo real, las heurísticas como Rate Monotonic Scheduling (RMS) o Earliest Deadline First (EDF) se usan para decidir el orden de ejecución de procesos con restricciones temporales.
- **Enrutamiento en redes de sensores inalámbricos (WSN):** Algoritmos como AODV o DSR implementan estrategias heurísticas para elegir rutas eficientes sin sobrecargar los nodos.
- **Compresión y codificación de datos:** Métodos heurísticos permiten una compresión eficiente sin necesidad de un procesamiento intensivo.
- **Optimización de consumo energético:** Heurísticas simples deciden cuándo apagar o activar módulos, o cómo ajustar la frecuencia del procesador para ahorrar energía.

4. Tipos de heurísticas empleadas

- **Constructivas:** Construyen una solución paso a paso, como el algoritmo de asignación de tareas basado en prioridad.
- **Greedy (avara):** Toman decisiones locales óptimas con la esperanza de encontrar una solución global aceptable. Ejemplo: heurística de selección de nodos para transmitir datos en una red.
- **Basadas en búsqueda local:** Modifican soluciones actuales en busca de mejoras. Muy útiles en ajuste de parámetros.
- **Aleatorias o probabilísticas:** Usan componentes de aleatoriedad para escapar de óptimos locales, como en recocido simulado.
- **Basadas en reglas:** Conjunto de condiciones y acciones diseñadas específicamente para un dominio.

5. Diseño de heurísticas eficaces

Para que una heurística sea útil en un sistema embebido debe:

- Ser **simple de implementar** en el hardware objetivo.
- Tener **tiempo de ejecución acotado y predecible**.
- Considerar **las restricciones particulares del sistema** (energía, latencia, memoria).
- Producir resultados **reproducibles y consistentes**.

El proceso de diseño incluye pruebas empíricas, afinación iterativa y, en ocasiones, la integración con otras técnicas (como aprendizaje automático o algoritmos evolutivos).

Ejemplos

- **Sistema de riego inteligente:** Una heurística decide si regar o no basándose en humedad del suelo, clima y nivel de batería, en lugar de usar un modelo predictivo complejo.
- **Selección de canal en redes inalámbricas embebidas:** En vez de calcular toda la interferencia, se usa una heurística que mide el nivel de ruido actual y escoge el canal con mejor promedio reciente.
- **Ajuste de brillo automático en pantallas embebidas:** Basado en condiciones de luz medidas y reglas simples (si luz < X, entonces brillo = alto).
- **Planificación de tareas en drones autónomos:** Asignar prioridades a tareas de navegación, comunicación y captura de imágenes mediante una tabla heurística que evalúa riesgo, urgencia y batería restante.

Conclusión

Las heurísticas son herramientas fundamentales para la solución eficiente de problemas en sistemas embebidos. Su capacidad para operar bajo condiciones restrictivas, adaptarse a cambios y ofrecer resultados aceptables en tiempos cortos las hace indispensables en contextos donde las soluciones óptimas exactas no son viables.

Al ser flexibles y de bajo costo computacional, las heurísticas se convierten en una opción ideal para tareas de planificación, control, optimización y comunicación dentro de estos sistemas. El éxito de su aplicación depende de un diseño cuidadoso, adaptado al dominio específico, y de una validación continua basada en pruebas reales.

Bibliografía

- ❖ Marwedel, P. (2011). *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*. Springer.
- ❖ Russell, S. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall.
- ❖ Vahid, F., & Givargis, T. (2001). *Embedded System Design: A Unified Hardware/Software Approach*. Wiley.
- ❖ Liu, J. W. S. (2000). *Real-Time Systems*. Prentice Hall.
- ❖ Zomaya, A. Y., & Teh, Y. H. (2001). Observations on using genetic algorithms for dynamic load-balancing. *IEEE Transactions on Parallel and Distributed Systems*.
- ❖ Buttazzo, G. C. (2011). *Hard Real-Time Computing Systems*. Springer.
- ❖ Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*.
- ❖ Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*.