

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 12:

Algoritmo genético

Algoritmo Genético en Sistemas Embebidos

Introducción

Los **algoritmos genéticos (GA, por sus siglas en inglés)** son una clase de metaheurísticas inspiradas en los principios de la evolución natural. Se utilizan para resolver problemas complejos de optimización donde no existen métodos exactos eficientes. Su capacidad para explorar grandes espacios de soluciones de forma paralela los hace ideales para múltiples aplicaciones en el diseño de **sistemas embebidos**, donde se busca optimizar parámetros como consumo energético, rendimiento, tamaño de código y eficiencia de hardware.

En sistemas embebidos, donde los recursos son limitados, los algoritmos genéticos se aplican en tareas como la asignación óptima de tareas, el ajuste de controladores, la selección de configuraciones hardware/software y el diseño evolutivo de arquitecturas digitales.

Desarrollo

1. Fundamentos del algoritmo genético

El algoritmo genético simula el proceso evolutivo de una población de individuos que se adapta a su entorno. Cada individuo representa una solución posible y se evalúa mediante una **función de aptitud (fitness)**. A través de selección, cruce y mutación, se generan nuevas generaciones que tienden a mejorar las soluciones anteriores.

2. Estructura del algoritmo

1. Codificación de soluciones

Las soluciones se representan como cadenas (generalmente binarias, pero

también enteras o reales), llamadas **cromosomas**.

2. Inicialización de la población

Se genera un conjunto inicial de soluciones aleatorias o heurísticas.

3. Evaluación de aptitud

Cada individuo se evalúa mediante una función objetivo $f(x)$ que refleja su calidad.

4. Selección

Se eligen los mejores individuos para reproducirse. Métodos comunes:

- Ruleta
- Torneo
- Selección elitista

5. Cruzamiento (crossover)

Dos individuos se combinan para producir descendientes mezclando partes de su información.

6. Mutación

Se altera aleatoriamente parte del cromosoma para mantener diversidad y explorar nuevas regiones.

7. Reemplazo

Se forma una nueva generación, reemplazando parcialmente o totalmente a la anterior.

8. Criterio de parada

Se detiene al alcanzar un número de generaciones o si no hay mejora durante cierto tiempo.

3. Ejemplo de función objetivo

En un sistema embebido que busca minimizar energía y tiempo de respuesta:

$$f(x) = \alpha \cdot E(x) + \beta \cdot T(x)$$

Donde:

- $E(x)$: energía consumida
- $T(x)$: tiempo de ejecución
- $\alpha\beta$: pesos que indican la prioridad de cada objetivo

4. Aplicaciones en sistemas embebidos

a) Asignación óptima de tareas en sistemas tiempo real

Se busca distribuir procesos en distintos núcleos o dispositivos de manera que se minimice el retardo total o se cumplan restricciones de tiempo.

b) Ajuste de parámetros en controladores PID embebidos

Se optimizan los valores de K_p , K_i , K_d para obtener el mejor desempeño en control de sistemas como motores, temperatura o posición.

c) Diseño de circuitos digitales evolutivos (evolvable hardware)

Utilizado para sintetizar configuraciones lógicas en FPGA o ASIC mediante evolución genética.

d) Selección automática de configuraciones de bajo consumo

Optimiza combinaciones de frecuencia, voltaje y módulos activos para maximizar duración de batería.

5. Ventajas y desventajas

Ventajas	Desventajas
Explora grandes espacios de búsqueda	Puede requerir muchos ciclos para converger
No necesita derivadas ni continuidad	Riesgo de converger prematuramente
Flexible y adaptable a diferentes problemas	Requiere ajuste fino de parámetros
Se puede paralelizar	Alto uso de memoria en algunas variantes

Ejemplo práctico (simplificado)

Problema: Ajuste de parámetros de un controlador embebido para un motor DC.

Representación:

- Cromosoma = $[K_p, K_i, K_d]$
- Población inicial: 20 individuos con valores aleatorios.

- Evaluación: función que mide error cuadrático medio (MSE) del sistema.
- Cruzamiento: promedio ponderado.
- Mutación: modificación aleatoria del 5% del valor.
- Parada: 50 generaciones o error < tolerancia.

Resultado esperado: Combinación de parámetros que minimiza el error de control con bajo tiempo de respuesta.

Conclusión

Los algoritmos genéticos son una herramienta eficaz para resolver problemas de optimización en sistemas embebidos, especialmente cuando el espacio de soluciones es amplio, no lineal o mal definido. Gracias a su capacidad de exploración global y adaptación, pueden aplicarse a tareas críticas de diseño, ajuste, planificación y ahorro energético.

Aunque requieren mayor esfuerzo computacional que heurísticas simples, su uso adecuado (o hibridado con otras técnicas como búsqueda local) puede marcar una gran diferencia en la eficiencia y rendimiento del sistema embebido.

Bibliografía

- ❖ Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.



Facultad de Ingeniería
Tampico

- ❖ 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.
- ❖ Deb, K. (2001). *Multi-objective Optimization using Evolutionary Algorithms*. Wiley.
- ❖ Coello, C. A. (2002). *Theoretical and Numerical Constraint-Handling Techniques Used with Evolutionary Algorithms: A Survey of the State of the Art*. Computer Methods in Applied Mechanics and Engineering.