Instituto Tecnológico de Costa Rica  
Unidad de Computación

**Curso:**Análisis de Algoritmos

**Profesora:**

Lorena Valerio Solís

**Análisis de complejidad algorítmica**

**Integrantes:**

Fabián Vargas Araya

Joseph Salas Rivas

**Sede:**

Local San Carlos

Junio 2025

# Introducción

En el campo de la computación, la resolución de Problemas de Satisfacción de Restricciones (CSP), como el clásico problema de las N-Reinas, representa un desafío fundamental para el diseño y análisis de algoritmos. El algoritmo de las N-Reinas “ha sido estudiado por más de un siglo desde su propueste en 1948 por el alemán Max Bezzel” (Ruiz, nd). Este proyecto aborda dicho problema, que consiste en posicionar N reinas en un tablero de ajedrez de NxN de tal forma que ninguna amenace a otra, implementando dos paradigmas algorítmicos fundamentalmente distintos para su solución y posterior comparación.

La primera estrategia es el Backtracking, un método de búsqueda exhaustiva y determinista. Este algoritmo construye una solución de forma incremental, explorando sistemáticamente el espacio de decisiones. Ante cada paso, si una restricción es violada (un "callejón sin salida"), la técnica realiza una poda de esa rama del árbol de búsqueda y retrocede a un punto de decisión anterior para explorar una alternativa. Gracias a su naturaleza exhaustiva, el backtracking garantiza encontrar una solución óptima o todas las soluciones posibles, aunque su costo computacional puede ser exponencial.

En contraparte, se implementa un Algoritmo Genético (AG), una heurística de búsqueda estocástica inspirada en los principios de la evolución biológica de Darwin. A diferencia del backtracking, un AG no opera sobre una única solución candidata, sino sobre una población de ellas. Cada solución, codificada como un cromosoma, es evaluada mediante una función de aptitud (fitness) que cuantifica su calidad. A través de generaciones sucesivas, la población evoluciona mediante la aplicación de operadores genéticos: la selección de los individuos más aptos, el cruce (crossover) para recombinar sus características y la mutación para introducir diversidad. Este enfoque permite a los AGs navegar paisajes de búsqueda complejos, con la capacidad de evitar óptimos locales y aproximarse a una solución globalmente óptima.

El objetivo de este trabajo es realizar un análisis comparativo riguroso de ambas estrategias, evaluando su desempeño en términos de eficiencia computacional (tiempo y memoria), calidad de la solución y escalabilidad. Ambas implementaciones se desarrollarán en Java, utilizando una única representación de datos y adhiriéndose a los principios de la Programación Orientada a Objetos para asegurar un código limpio, modular y mantenible.

# Análisis del Problema

…

# Solución del Problema

…

# Análisis de Resultados

…

# Conclusiones

…

# Recomendaciones

…

# Literatura Citada

MathWorks - Algoritmo genético:

Es una introducción conceptual y de alto nivel. Su mayor aporte es la tabla que contrasta un "Algoritmo clásico" con un "Algoritmo genético".

Proporciona la dicotomía fundamental para la introducción: Backtracking (clásico) es determinista y explora un único punto a la vez en una secuencia, mientras que el Genético (evolutivo) es estocástico (aleatorio) y opera sobre una población de puntos simultáneamente. Esta es una idea central para comparar ambas estrategias.

GeeksforGeeks - Introducción al Backtracking:

Es un tutorial detallado que formaliza el concepto de backtracking. Define terminología clave como "solución parcial", "espacio de decisión", "dead end" (callejón sin salida) y "poda".Es la fuente principal para definir Backtracking de manera formal. En lugar de decir "probar diferentes caminos", podemos usar su terminología: "construir una solución de forma incremental" y "realizar poda del espacio de búsqueda" cuando una restricción se viola. Esto eleva la precisión técnica de la descripción.

Medium - Retroceso en profundidad con problemas de LeetCode:

Este artículo conecta el backtracking directamente con la clase de problemas que resuelve: Problemas de Satisfacción de Restricciones (CSP), mencionando explícitamente el de las N-Reinas. Nos permite enmarcar el problema de las N-Reinas no como un simple acertijo, sino como un ejemplo canónico de un CSP. Usar este término en la introducción le da un peso académico y contextualiza el problema dentro de un campo de estudio formal de la IA y los algoritmos.

UTN (GIAR) y U. de Sevilla (CS.US.ES) - Algoritmos Genéticos (PDFs/Blog):

Ambos son documentos académicos y muy completos. Introducen la terminología formal de los AGs: cromosoma (representación de la solución), función de aptitud (fitness), operadores genéticos (selección, cruce, mutación) y la base teórica como la "Hipótesis de los bloques constructivos".Son cruciales para describir el algoritmo genético con la terminología correcta y demostrar un entendimiento profundo. En lugar de "mejores candidatos", hablaremos de individuos con mayor "aptitud". En vez de "cruce", usaremos los términos formales de los operadores.

Conogasi - Algoritmos genéticos:

Un artículo de divulgación que explica muy bien la motivación de los AGs: la búsqueda de óptimos globales en lugar de quedarse atascado en óptimos locales.

Aporta el "porqué" de los algoritmos genéticos. Nos permite explicar que, a diferencia de otras técnicas que pueden quedar atrapadas en una solución "buena pero no la mejor" (óptimo local), los AGs tienen un mecanismo para explorar el espacio de búsqueda de forma más amplia, aspirando a una solución globalmente mejor.

GeeksforGeeks. (2024, June 24). *Introduction to backtracking*. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-to-backtracking-2/>

Ruiz, I. O. (n.d.). *El Problema de Las N-Reinas Con Algoritmos Genéticos*. Scribd. <https://www.scribd.com/doc/291059404/El-Problema-de-Las-N-Reinas-Con-Algoritmos-Geneticos>

(Ruiz, n.d.)