

Informe Técnico: IA para el Juego de Hex

John García Muñoz C-311

11 de abril de 2025

1. Heurísticas de Evaluación

1.1. Distancia de Dijkstra Modificada

Esta heurística calcula el camino de costo mínimo que un jugador necesita para conectar sus lados del tablero. Asigna costos diferenciados:

- **Costo 0:** Para casillas ocupadas por el jugador
- **Costo 1:** Para casillas vacías
- **Costo 10:** Para casillas del oponente (bloqueos)

La diferencia entre las distancias de ambos jugadores determina la ventaja posicional.

1.2. Detección de Puentes y Conexiones Forzadas

Un puente es una estructura donde dos fichas están separadas por dos casillas vacías adyacentes, tal que estas dos se tienen "vértices adyacentes", permitiendo una conexión en un solo movimiento, de tal forma que se puede considerar a estas como conectadas aunque no lo estén directamente, pues esta es forzada.

Esta técnica explota la naturaleza del Hex, donde se premian las conexiones tácticas, en particular generar ocasiones de conexiones indirectas forzadas, ahorrando movimientos.

1.3. Control de Áreas Estratégicas

El centro del tablero y las regiones cercanas a los bordes objetivos tienen gran valor teórico al inicio del juego. La heurística asigna puntuaciones basadas en la distancia hacia el centro de las casillas. Esta se encuentra ponderada para que sea un valor mínimo, de tal forma que solo sea apreciable al comienzo, cuando el resto de las heurísticas son despreciables en comparación. Garantiza que la primera jugada sea al centro.

1.4. Heurística de Bloqueo

Penaliza las casillas que se encuentran bloqueadas por casillas del adversario, por convención, se hace calculando la cantidad de vecinos oponentes menos la cantidad de vecinos aliados, si esta diferencia es mayor o igual que 2, consideramos bloqueada esta casilla, no interesa mover ahí, y se penaliza.

2. Optimizaciones

2.1. Tabla de Transposición con Hash Zobrist

Esta optimización utiliza el **hash Zobrist**, un método para identificar de forma única cada estado del tablero mediante operaciones **XOR**. Antes de iniciar el juego, se genera una tabla de números aleatorios de 64 bits para cada posible combinación de (fila, columna, jugador). Al realizar un movimiento, el hash se actualiza combinando (**XOR**) los valores correspondientes a la casilla modificada.

2.2. Gestión Adaptativa de la Profundidad

Para maximizar la eficiencia dentro del límite de tiempo, la IA ajusta dinámicamente la profundidad de búsqueda. Utiliza un registro histórico del tiempo invertido en cada nivel de profundidad (ej: 0.5 segundos para profundidad 3, 2 segundos para profundidad 4). Con este dato, estima cuántos niveles adicionales puede explorar antes de agotar el tiempo.

2.3. Ordenamiento de Movimientos

El ordenamiento estratégico de movimientos acelera la poda alfa-beta al explorar primero las jugadas más prometedoras. Se priorizan casillas en zonas críticas: el **centro** del tablero (por su valor estratégico), **puentes** (para crear conexiones propias o bloquear las del rival) y **bordes** (claves para la conexión final).