



Universidad Peruana de Ciencias Aplicadas

Facultad de Ingeniería

Ciclo 04

Nombre del curso:
Complejidad Algorítmica

Sección:
CC42

Nombre del profesor:

"Informe de Trabajo Parcial"

Relación de integrantes:

Rojas Cuadros Fabian Marcelo - U202218498
Su Caletti Eddo - U20221A390
Alexander Fernandez Garfias - U202019498

Índice de contenido

1. Descripción del problema:.....	3
2. Descripción del conjunto de datos:.....	3
2.1 Estructura del dataset:(grafos).....	4
3. Propuesta:.....	5
3.1 Objetivo:.....	5
3.2 Tecnicas y metodologia a utilizar:.....	5
3.3 Metodología:.....	5
4. Bibliografías:.....	6

1. Descripción del problema:

Este proyecto se centra en el desarrollo de un código para simular partidas del juego de damas, en el cual los movimientos de las piezas se realizarán de manera aleatoria. Los movimientos se determinarán en función de la estructura del tablero, el tamaño del mismo y las piezas involucradas, así como el número de jugadores. El objetivo es evaluar cómo la disposición inicial y las condiciones del juego influyen en los movimientos posibles, utilizando técnicas para encontrar soluciones óptimas a partir de movimientos generados aleatoriamente.

Según Gupta et al. (2021), el juego de damas puede modelarse eficazmente utilizando un grafo, donde cada casilla del tablero representa un nodo y los movimientos legales entre las casillas se representan como arcos del grafo. Este enfoque permite aplicar estrategias de búsqueda heurística para optimizar las decisiones del programa, explorando diferentes configuraciones del tablero y movimientos posibles de manera sistemática.

En este contexto, las técnicas de búsqueda heurística son fundamentales para resolver problemas complejos como el juego de damas, proporcionando métodos eficientes para explorar y evaluar opciones de juego (Allis, 1994; López, 2022). Estas estrategias permiten al programa adaptarse dinámicamente a las condiciones cambiantes del juego, mejorando la calidad de las decisiones tomadas durante la simulación de partidas.

2. Descripción del conjunto de datos:

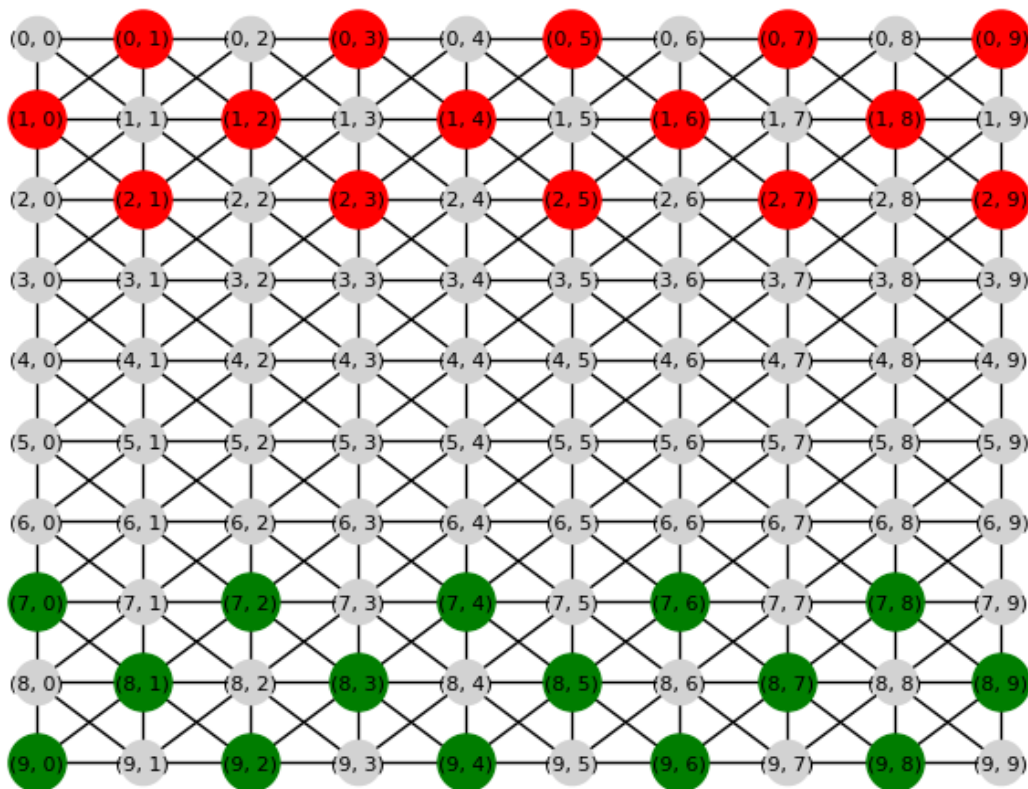
El conjunto de datos utilizado para la simulación se genera a partir de la estructura del tablero de damas, modelado como un grafo donde las casillas representan los nodos y los movimientos permitidos entre casillas adyacentes o a través de saltos representan los arcos. Para la gestión y procesamiento de estos datos se utilizan las siguientes herramientas y librerías:

- **Python:** Lenguaje de programación para desarrollar el simulador.
- **Librería networkx:** Permite crear y manipular el grafo que representa el tablero y las conexiones entre las casillas.
- **Librería numpy:** Utilizada para manejar las matrices de adyacencia que representan las conexiones entre los nodos.
- **Librería matplotlib:** Empleada para la visualización gráfica del tablero y las posiciones de las piezas.

2.1 Estructura del dataset:(grafos)

- **Nodos:** Cada nodo representa una casilla del tablero de damas y contiene información sobre su estado (ocupada o libre).
- **Aristas:** Representan los movimientos legales entre las casillas, incluyendo posibles saltos.
- **Matriz de adyacencia:** Se genera dinámicamente una matriz de adyacencia de 121 nodos (casillas) que representa las conexiones de movimiento en el tablero.

Estas herramientas permiten simular el comportamiento del juego y registrar los movimientos generados para su análisis posterior.



La imagen muestra un modelo gráfico del tablero de 10x10 utilizado en el juego. Cada casilla está representada como un nodo conectado con sus adyacentes mediante aristas, simulando las posibles rutas de movimiento. Las fichas del jugador 1 aparecen en rojo y las del jugador 2 en verde. Esta visualización referencial permite entender cómo se organizan las posiciones iniciales de las fichas y sus conexiones en el tablero.

3. Propuesta:

3.1 Objetivo:

El objetivo de este proyecto es desarrollar un simulador del juego de damas utilizando técnicas algorítmicas y modelos de grafos. La propuesta se basa en modelar el tablero como un grafo donde cada nodo representa una casilla y los arcos representan los movimientos permitidos entre casillas. La simulación de los movimientos de las piezas se realizará de manera aleatoria, utilizando un enfoque de búsqueda basado en técnicas heurísticas para encontrar las rutas óptimas o posibles caminos para cada jugador.

3.2 Tecnicas y metodologia a utilizar:

Backtracking:

Se utilizará para explorar todas las soluciones posibles en el juego de damas mediante búsqueda por ensayo y error. Este enfoque generará diferentes configuraciones para las piezas, descartando aquellas que no cumplan con las reglas o que no sean óptimas. A través del retroceso, el algoritmo permitirá identificar movimientos válidos y optimizar la toma de decisiones en cada turno, asegurando que se consideren todas las rutas posibles.

Programación Dinámica:

Se implementará para resolver el problema dividiendo la situación del juego en subproblemas más pequeños y almacenando sus soluciones. Este enfoque será clave para optimizar las decisiones en el juego, permitiendo evaluar múltiples configuraciones y determinar el mejor movimiento en función de las posiciones actuales de las piezas. Almacenar las soluciones de subproblemas evitará la re-evaluación y mejorará la eficiencia del proceso de toma de decisiones.

3.3 Metodología:

- **Modelado del grafo:** El tablero se representará como un grafo en el que cada casilla es un nodo y las conexiones entre casillas representan los movimientos legales de las piezas.

- Simulación de movimientos: Los movimientos se generarán de manera aleatoria, evaluando las opciones posibles en cada turno de acuerdo con las reglas del juego.
- Evaluación de movimientos: Se emplearán técnicas heurísticas para analizar las posibles configuraciones y encontrar las mejores jugadas posibles en cada turno.

El simulador permitirá visualizar los movimientos de las piezas a lo largo de la partida y analizar cómo las configuraciones iniciales influyen en el desarrollo del juego, utilizando las herramientas y librerías mencionadas previamente.

4. Bibliografías:

Gupta, P., Nagrath, V., & Preeti, N. (2021). Checkers-AI. En *Deep Learning in Gaming and Animations* (pp. 1-18). CRC Press.

<https://www.taylorfrancis.com/chapters/edit/10.1201/9781003231530-1/checkers-ai-priyanshi-gupta-vividha-preeti-nagrath>

Building a Checkers Gaming Agent Using Deep Q-Learning. (n.d.).

<https://blog.paperspace.com/building-a-checkers-gaming-agent-using-neural-networks-and-reinforcement-learning/>

GeeksforGeeks. (s.f.). *Introducción a los algoritmos*. Recuperado el 22 de septiembre de 2024, de <https://www.geeksforgeeks.org/introduction-to-algorithms/>

Khan Academy. (s.f.). *Algoritmos*. Recuperado el 22 de septiembre de 2024, de <https://www.khanacademy.org/computing/computer-science/algorithms>

Codecademy. (s.f.). *Aprende algoritmos y estructuras de datos*. Recuperado el 22 de septiembre de 2024, de <https://www.codecademy.com/learn/paths/algorithms>