Tarea 1: Búsqueda y Resolución de Problemas

Asignación Óptima de Recursos en Centro de Distribución

Curso: Inteligencia Artificial
Facultad de Ingeniería y Ciencias
Universidad Adolfo Ibáñez

Profesores: Camilo Ramirez, Mauricio Figueroa, Mauricio A. Valle
Fecha Distribución: [Fecha a definir]
Fecha Entrega: [Fecha a definir]

Objetivo

Entender y aplicar conceptos fundamentales de búsqueda en espacios de estados para la resolución de problemas de asignación de recursos, comparando el desempeño de diferentes algoritmos de búsqueda (BFS, DFS, UCS) en términos de optimalidad, complejidad temporal y espacial.

1. Contexto

GMB Solutions es una empresa de distribución que opera un centro automatizado de almacenamiento y distribución. El centro cuenta con múltiples estaciones de trabajo especializadas donde se procesan diferentes tipos de productos antes de ser enviados a los clientes. Cada estación tiene capacidades específicas y costos operacionales diferentes.

La empresa maneja dos tipos principales de productos:

- **Productos Farmacéuticos** (requieren estaciones con certificación de temperatura)
- **Productos Alimentarios** (requieren estaciones con certificación sanitaria)

El centro de distribución está organizado en una red de 23 estaciones interconectadas (E1 a E23), donde cada producto individual debe encontrar su propia ruta desde la estación de entrada (E1) hasta la estación de salida (E23).

2. Ejemplo Ilustrativo

Para comprender mejor el problema, consideremos un sistema simplificado:

2.1 Configuración Simplificada

Estaciones:

- **E1**: Entrada (ambas certificaciones, costo $0)
- **E2**: Temperatura (para farmacéuticos, costo $10)
- **E3**: Sanitaria (para alimentarios, costo $8)
- **E4**: Salida (ambas certificaciones, costo $0)

Conexiones:

E1 → E2 → E4

E1 → E3 → E4

2.2 Estados del Sistema

Estado Inicial:

python

```
{
  'E1': ['P1', 'P2'],  # Ambos productos en entrada
  'E2': [], 'E3': [], 'E4': [],
  'procesados': []    # Productos que llegaron a E4
}
```

Estado Objetivo:

python

```
{
  'E1': [], 'E2': [], 'E3': [], 'E4': [],
  'procesados': ['P1', 'P2']  # Ambos en salida
}
```

2.3 Secuencia de Acciones Ejemplo

1. **Mover P1:** E1 → E2 (costo $10)
2. **Mover P1:** E2 → E4 (producto procesado)

3. **Mover P2:** E1 → E3 (costo $8)
4. **Mover P2:** E3 → E4 (producto procesado)

Costo total: $18

3. Problema

Cada producto individual (P1, P2) debe encontrar su propia ruta desde la estación de entrada E1 hasta la estación de salida E23. La condición de "producto procesado" solo ocurre cuando el producto llega a la estación E23. Cada vez que un producto pasa por una estación, se aplica el costo de procesamiento de dicha estación.

3.1 Formulación del Problema

Deberán formular este problema como un **problema de búsqueda en espacios de estados**, identificando claramente:

1. **Estados**: Configuración actual del sistema (qué productos están en qué estaciones y cuáles han sido procesados)
2. **Estado Inicial**: Todos los productos en la estación de entrada, ninguno procesado
3. **Estado Objetivo**: Todos los productos han sido procesados exitosamente
4. **Acciones**:
    - Movimiento de productos individuales entre estaciones conectadas
    - Procesamiento de productos en estaciones con certificaciones apropiadas
5. **Función de Transición**: Reglas que determinan los movimientos y procesamientos válidos
6. **Función de Costo**: Costo acumulado de procesamiento de productos
7. **Espacio de Estados**: Conjunto de todas las configuraciones posibles del sistema

3.2 Datos Proporcionados

Se proporcionan los siguientes archivos:

- `estaciones.npy`: Información de las estaciones
- `conexiones.npy`: Conectividad entre estaciones

3.3 Estructura de Datos

Tabla 1: Estructura del archivo estaciones.npy

Atributo	Descripción	Tipo
id_estacion	Identificador único de la estación	String
certificaciones	Lista de certificaciones que posee la estación	Array
costo_procesamiento	Costo por producto procesado en esta estación	Float

Tabla 2: Estructura del archivo conexiones.npy

Atributo	Descripción	Tipo
estacion_origen	ID de estación de origen	String
estacion_destino	ID de estación de destino	String

3.5 Detalles de los Datos

Distribución de estaciones:

- **E1**: Entrada (certificaciones: temperatura + sanitaria, costo: $0)
- **E2-E11**: 10 estaciones solo con certificación "temperatura" (costos: $8.9 - $14.8)
- **E12-E22**: 11 estaciones solo con certificación "sanitaria" (costos: $6.9 - $11.7)
- **E23**: Salida (certificaciones: temperatura + sanitaria, costo: $0)

Conectividad: La red está diseñada para tener múltiples caminos alternativos para ambos tipos de productos, con aproximadamente 47 conexiones totales.

4. Implementación Requerida

4.1 Algoritmos de Búsqueda

Implementar los siguientes algoritmos desde cero (no usar librerías como NetworkX):

1. **Breadth-First Search (BFS)**
2. **Depth-First Search (DFS)**
3. **Uniform Cost Search (UCS)**

4.2 Análisis de Complejidad

Para cada algoritmo implementado, deberán analizar:

- **Complejidad Temporal**: O(?) en función del factor de ramificación (b) y profundidad (d)
- **Complejidad Espacial**: O(?)
- **Optimalidad**: ¿Garantiza encontrar la solución óptima?
- **Completitud**: ¿Garantiza encontrar una solución si existe?

4.3 Comparación Experimental

Ejecutar los tres algoritmos sobre el mismo problema y comparar:

- Costo de la solución encontrada
- Número de nodos explorados
- Longitud del camino solución (número de acciones)

5. Entregables

5.1 Notebook de Google Colab

Un notebook ejecutable que contenga:

1. **Carga y exploración de datos**
2. **Formulación formal del problema**
3. **Implementación de los tres algoritmos**
4. **Función de visualización de la solución**
5. **Experimentos comparativos**
6. **Análisis de resultados**

5.2 Reporte Escrito (en Markdown dentro del notebook)

El reporte debe incluir las siguientes secciones:

5.2.1 Formulación del Problema

- Definición formal de estados, acciones, objetivos
- Justificación de la representación elegida
- Análisis del espacio de estados y factor de ramificación

5.2.2 Explicación de las decisiones de implementación

5.2.3 Resultados Experimentales

- Tablas comparativas de rendimiento
- Gráficos de visualización de soluciones
- Análisis de trade-offs entre algoritmos

5.2.4 Discusión y Conclusiones

- Interpretación de resultados
- Recomendaciones sobre cuándo usar cada algoritmo
- Limitaciones identificadas y posibles mejoras

5.2.5 Contribución Individual

- Descripción clara del trabajo realizado por cada integrante
- Distribución de responsabilidades

5.3 Visualización Requerida

- **Grafo de la red de estaciones** con conexiones
- **Comparación visual** de las soluciones encontradas por cada algoritmo

6. Consideraciones Técnicas

6.1 Restricciones del Problema

* Productos farmacéuticos solo pueden moverse por estaciones con certificación "temperatura

- Productos alimentarios solo pueden moverse por estaciones con certificación "sanitaria"
- E1 y E23 tienen ambas certificaciones
- El movimiento solo es válido si existe conexión directa entre estaciones
- Un producto solo puede estar en una estación a la vez

6.2 Condiciones de Entrega

- **Equipos**: Máximo 3 integrantes
- **Formato**: Un archivo `grupoX_tarea2_IA.ipynb` donde X es el identificador del grupo
- **Plataforma**: Google Colab (debe ejecutarse sin errores)
- **Penalización**: 0.5 puntos por día de atraso

6.3 Criterios de Evaluación

- **Correctitud de la implementación** (25%)
- **Calidad de la formulación del problema** (20%)
- **Análisis comparativo de algoritmos** (20%)
- **Visualización y presentación** (15%)
- **Calidad del reporte y documentación** (15%)
- **Contribución individual clara** (5%)