Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-5437 Inteligencia Artificial I
Trimestre Abril - Julio 2023

Informe Proyecto I

Estudiantes:

Br. Gamboa, Roberto

Br. Bandez, Jesús

Carnés:

16-10394

17-10046

Profesor

Carlos Infante

13 de junio de 2023

Índice

Introducción

La inteligencia artificial ha revolucionado numerosos campos y ha desempeñado un papel fundamental en el desarrollo de nuevas tecnologías que facilitan la automatización de tareas complejas. Uno de los conceptos fundamentales en el ámbito de la inteligencia artificial es la representación de problemas como espacios de estados.

La representación de problemas como espacios de estados es una técnica crucial en la resolución de problemas mediante algoritmos de búsqueda y planificación. Esta representación proporciona una estructura formal para describir y analizar problemas, permitiendo a los sistemas de inteligencia artificial examinar diferentes combinaciones de acciones y estados para determinar la secuencia óptima que lleve a la solución del problema. Al expresar los problemas como espacios de estados, se pueden aplicar algoritmos de búsqueda y planificación que exploran sistemáticamente las posibles soluciones y determinan la más adecuada.

En este trabajo se estudiará la representación de problemas mediante espacios de estados con problemas conocidos como los Npuzzles, el cubo de Rubik, la Torre de Hanoi y los Top Spin puzzles, además de aplicar heurísticas conocidas, algoritmos informados y árboles de búsqueda para explorar de qué manera se resuelven dichos problemas utilizando la API PSVN, especializada en resolver este tipo de problemas.

Decisiones de diseño e implementación

Para abordar el presente proyecto, inicialmente se utilizaron las representaciones de espacio de estados proporcionadas para los siguientes problemas:

- N-puzzles: 15-puzzle y 24-puzzle
- Cubo de Rubik: 3x3x3
- Top spin: 12-4, 14-4, y 17-4
- Torre de Hanoi con 4 astas: 12, 14, y 18 discos

Las cuales se pueden encontrar en el subdirectorio Definitions correspondiente a cada problema, bajo el directorio Problems. En cada uno de estos directorios además se encuentran las definiciones de las abstracciones de cada problema, obtenidas utilizando el abstractor suministrado junto con el software PSVN, las cuales son utilizadas para aplicar las distintas heurísticas y PDBs para encontrar la solución de estos problemas y diversos archivos .txt que contienen distintos estados iniciales y estados objetivos para cada problema estudiado.

Dentro del directorio correspondiente se encuentran también las carpetas Heuristics, que contiene los archivos Makefile para compilar cada problema y scripts para ejecutar las heurísticas correspondientes; SearchTree que contiene los archivos para ejecutar los árboles de búsqueda de cada problema, y Solver, que contiene los archivos para crear y ejecutar las PDBs u otras heurísticas correspondientes a cada problema

Adicionalmente se realizaron implementaciones de los algoritmos BFS, IDA* y A* para poder ejecutar cada parte del proyecto, dichas implementaciones se encuentran dentro del subdirectorio Global del directorio Problems

Árboles de búsqueda

Para cada problema se partió desde el estado objetivo, y aplicando reglas inversas se generaron dos árboles de búsqueda por problema, uno utilizando eliminación parcial de duplicados (poda de ancestros) y el otro sin utilizar eliminación de duplicados. Los resultados obtenidos se presentan en la siguiente tabla, obtenidos tras 10 minutos de ejecución del algoritmo BFS para cada problema en los casos que fue posible, o hasta que se agotó la memoria del computador donde no. Los resultados para cada problema se presentan en la siguiente tabla:

Problema	**BFS sin eliminación de duplicados**	**BFS + poda de ancestros**
15-Puzzle	Profundidad alcanzada: 16 Nodos expandidos: 33.869.011	Profundidad alcanzada: 24 Nodos expandidos: 76.610.933
24-Puzzle	Profundidad alcanzada: 15 Nodos expandidos: 19.546.809	Profundidad alcanzada: 21 Nodos expandidos: 46.732.751
Torre de Hanoi 12 Discos	Profundidad alcanzada: 11 Nodos expandidos: 13.569.751	Profundidad alcanzada: 11 Nodos expandidos: 6.432.331
Torre de Hanoi 14 Discos	Profundidad alcanzada: 11 Nodos expandidos: 13.569.751	Profundidad alcanzada: 11 Nodos expandidos: 6.432.331
Torre de Hanoi 18 Discos	Profundidad alcanzada: 11 Nodos expandidos: 13.569.751	Profundidad alcanzada: 11 Nodos expandidos: 6.432.331
Cubo de Rubik 3x3	Profundidad alcanzada: 6 Nodos expandidos: 2.000.719	Profundidad alcanzada: 7 Nodos expandidos: 8.331.112
Top Spin 12-4	Profundidad alcanzada: 7 Nodos expandidos: 3.257.437	Profundidad alcanzada: 8 Nodos expandidos: 3.579.455
Top Spin 14-4	Profundidad alcanzada: 7 Nodos expandidos: 8.108.731	Profundidad alcanzada: 8 Nodos expandidos: 6.594.815
Top Spin 17-4	Profundidad alcanzada: 6 Nodos expandidos: 1.508.598	Profundidad alcanzada: 8 Nodos expandidos: 14.388.178

De los resultados presentados en la tabla se pueden extraer las siguientes conclusiones:

- Al aplicar poda de ancestros se pueden explorar muchos mas estados y en algunos casos llegar a una profundidad mayor al tener menos memoria ocupada en estados repetidos

- Mientras más complejo es la representación de espacio de estados de un problema, se logra explorar menos estados ya que estos cuentan con una mayor cantidad de reglas que se deben considerar en cada estado y a su vez se consumen muchos más recursos del equipo
- Se presenta un factor de ramificación exponencial, especialmente acentuado en los problemas más complejos

El detalle de la cantidad de nodos a cada profundidad por cada problema se puede encontrar en el directorio de cada problema, bajo el subdirectorio SearchTree en archivos .txt

Heurísticas PDB

Para los problemas NPuzzle se realizaron heurísticas PDB aditivas, de la forma 6-5-5 para el 15Puzzle y 5-5-5-5-5 para el 24Puzzle, las mismas, asi como las PDBs de cada problema pueden ser compiladas mediante el script ./create_pdbs.sh ubicado en la carpeta Heuristics de cada problema. Para los otros problemas se implementaros diversas PDBs no aditivas, las cuales son posteriormente utilizadas por los algoritmos informados para así hallar la solución a los problemas. En el caso particular del Cubo de Rubik se realizaron en primera instancia 2 PDBs, una PDB para las esquinas y otra PDB para las demás piezas excluyendo las esquinas; pero debido a la magnitud de las mismas se busco dividirlas es PDBs mas pequeñas, una que usara las esquinas de la parte superior del cubo, otra para las esquinas de parte inferior del cubo y otra para las demás piezas pero no se logró compilar dichas PDBs para poder ser usadas con los algoritmos.

Algoritmos informados

Para utilizar las PDBs generadas como se menciona en el apartado anterior, se realizaron implementaciones de los algoritmos IDA* y A* basados en los ejemplos proporcionados adaptados para el software PSVN, sin embargo, la implementación del algoritmo A* no funciona correctamente y no logra llegar a una solución en ninguno de los problemas.

En cada problema, a excepcion de los NPuzzle, se probaron las heurísticas junto con el algoritmo IDA* al para resolver diversos problemas de dificultad fácil, media y difícil. Los

resultados para cada problema se detallan a continuación, la ejecución del algoritmo se detiene al conseguir una solución o al pasar 10 minutos

Npuzzles

Para los Npuzzles se utilizó el máximo del valor obtenido entre todas las heurísticas PDB usadas, lo cual puede haber influido en los resultados obtenidos. Los resultados se muestran a continuación:

Problema	Cant. problemas	Cant. problemas resueltos	Max. profundidad	Nodos expandidos	Tiempo ejecución
15Puzzle	2	1	17	28.000	10 minutos
24Puzzle	2	1	23	414.000.000	10 minutos

Torre de Hanoi

Para la resolución de la Torre de Hanoi se utilizó el como heurística PDB el valor máximo obtenido entre diferentes PDBs al aplicar el algoritmo IDA* y se apreció que al incluir varias PDBs se acelera notablemente la velocidad a la que se llega a una solución. Los resultado para cada variación del problema se presentan a continuación:

12 discos

Dificultad	Canti. problemas	Cant. problemas resueltos	Máx. profundidad	Nodos expandidos	Tiempo de ejecución
Fácil	10	10	2	21	0,1 s
Media	10	10	40	285	0,1 s
Difícil	1	0	45	20.000.000	10 minutos

14 discos

Dificultad	Cant. problemas	Cant. problemas resueltos	Máx.profundidad	Nodos expandidos	Tiempo de ejecución
Fácil	10	10	5	27	0,9 s
Media	10	10	35	254	0,1 s
Difícil	1	0	54	<629.000.000	10 minutos

18 discos

Dificultad	Cant. problemas	Cant. problemas resueltos	Máx. profundidad	Nodos expandidos	Tiempo de ejecución
Fácil	10	10	4	25	1,28 s
Media	10	10	36	267	0,2 s
Difícil	1	0	55	272.000.000	10 minutos

Top Spin

Para el problema Top Spin, de manera similar a la Torre de Hanoi se utilizó como Heurística PDB el máximo de diversas heurísticas al usar el algoritmo IDA*. Los resultados se muestran a continuación:

12-4

Dificultad	Cant. problemas	Cant. problemas resueltos	Máx. profundidad	Nodos expandidos	Tiempo de ejecución
Fácil	10	10	5	30	0,008 s
Media	10	10	11	110.000	0,13 s
Difícil	10	10	11	190.000	0,13 s

14-4

Dificultad	Cant. problemas	Cant. problemas resueltos	Máx. profundidad	Nodos expandidos	Tiempo de ejecución
Fácil	10	10	5	50	0,028 s
Media	10	10	13	1.000.000	9,5 s
Difícil	10	10	12	3.000.000	1,52 s

17-4

Dificultad	Cant. problemas	Cant. problemas resueltos	Máx. profundidad	Nodos expandidos	Tiempo de ejecución
Fácil	10	10	5	70	0,7 s
Media	8	7	15	1.000.000.000	10 minutos
Difícil	1	1	15	5.659.000.000	10 minutos

Conclusión

La importancia de la representación de problemas como espacios de estados radica en su capacidad para abordar problemas complejos de manera sistemática y eficiente. Al proporcionar una estructura formal, se simplifica la tarea de modelar y resolver problemas complejos en una amplia gama de dominios, desde la logística hasta la robótica y el aprendizaje automático.

En el presente proyecto, luego de manipular las representaciones en espacio de estados de diversos problemas, trabajar con ellas, realizar abstracciones de las mismas y encontrar sus soluciones mediante el software PSVN se pudo presenciar de primera mano la efectividad de esta representación para encontrar la solución de una manera práctica.