****

**Tarea 3.4 Búsqueda Heurística**

**Inteligencia artificial**

Héctor Javier Delgado Neveu

5M - Desarrollo de Software

Registro 19310176

*24 de noviembre de 2021*

**INTRODUCCIÓN**

Las instrucciones de la tarea de investigación sobre la búsqueda heurística fueron las siguientes.

Realizar una investigación sobre la búsqueda Heurística, definiciones, aplicaciones de la heurística, algoritmos y problemas (A\*, distancia Manhattan, El problema del viajante, ascensión a la colina, Algoritmo voraz, etc.) más comúnmente utilizados en este tipo de búsquedas.

Realice un reporte que incluya:

Portada

índice

Introducción

Desarrollo, con descripciones gráficas de los algoritmos y problemas,

Conclusiones

Referencias bibliográficas.

**DESARROLLO**

**Búsqueda Heurística**

Se les conoce a heurísticas a “criterios, métodos o principios para decidir cuál de entre varias acciones promete ser la mejor para alcanzar una determinada meta”. Estas pueden servir como guías en la generación de árboles de búsqueda para conocer qué tan prometedor es un estado. Su uso hace que las búsquedas sean muchísimo más rápidas en comparación a búsquedas simples.

Este concepto tiene varias aplicaciones en la informática como en la optimización de programas, en programación web, ingeniería forense para determinar causas, problemas de diseño, cálculo de recursos necesarios, resolución más rápida de problemas conocidos, entre otras más.

**Algoritmos y Problemas**

Una función heurística retorna lo que es una estimación del coste del camino más corto para llegar al nodo solución en referencia a su posición actual del recorrido. Esta debe de ser admisible (no sobrestimar el coste real, siempre debe ser un valor menor) y ser consistente.

* Algoritmo Voraz: también conocida como *Greedy Search*, es un algoritmo de búsqueda enfocada en elegir siempre la opción óptima en cada paso del recorrido que hace con el propósito de obtener una solución general óptima. Usualmente es aplicado en problemas de optimización. Sus ventajas son que admite costes variables de acciones, reducción de complejidad y no recorre caminos inútiles. Entre sus desventajas esta que no es completo (porque puede terminar en callejones sin salida) y el hecho de que no es óptimo (porque no toma en cuenta los costes de acciones).

Text

Description automatically generated

* Algoritmo A\*: este algoritmo de búsqueda sí toma en cuenta el coste del camino recorrido y el coste de la heurística. Este sí es tanto completo como óptimo, pues se asegura siempre de encontrar una solución y esta siempre es la óptima. Su tiempo de ejecución por lo tanto es mayor, pero tiene oportunidades de ser reducido dependiendo del contexto y heurística utilizada. Entre otras ventajas y desventajas notables se encuentran que no recorre caminos inútiles y que se pueden seguir dando complejidades exponenciales en el espacio respectivamente.

Text

Description automatically generated

* Distancia Manhattan: “también conocida como taxicab metric, rectilinear distance o L1 distance, define la distancia entre dos puntos p y q como el sumatorio de las diferencias absolutas entre cada dimensión”. Se asigna a cada recuadro de la siguiente rompecabeza el valor de la suma de las distancias horizontales y verticales dependiendo de su posición.

A picture containing calendar

Description automatically generated

La función heurística en este caso sería la “suma de las distancias de cada una de las casillas (excluyendo la que se encuentra vacía)”.

* Diagram, radar chart

  Description automatically generatedEl problema del viajante: este problema plantea la siguiente situación: un viajante se encuentra en una ciudad específica y busca viajar a otra ciudad por la mejor ruta posible. Las ciudades que colindan a estas otras dos “están unidas por carreteras; se dispone de un mapa con la disposición de las provincias y sus "coordenadas" en kilómetros respecto al "centro" (por ejemplo, Madrid, con coordenadas (0,0))”. La heurística en este caso asignaría a cada estado el valor que sea equivalente a la distancia en línea recta hacia el estado objetivo. Se elige la ciudad siguiente cuando la suma de la distancia a la ciudad actual + la distancia aérea a la meta sea menor.

**CONCLUSIÓN**

Las heurísticas agregan otra capa de complejidad y de optimización a los algoritmos de búsqueda previamente investigados en la materia. No cabe duda de que tienen una utilidad crítica en una serie de situaciones y que su uso es prácticamente obligatorio si se quiere obtener los mejores resultados en optimización, tiempo y eficiencia. Entre los diferentes algoritmos en los que estas son empleadas varían su participación y su complejidad, lo cual permite comparar sus usos, tiempos y efectividad. De igual manera, cada algoritmo está diseñado para diferentes niveles de complejidad y necesidad, depende completamente del contexto si se requiere de uno u otro.

**Referencias bibliográficas**

Malagón, C. (s.f.). Búsqueda heurística. Consultado el 24 de noviembre de 2021. Disponible en <https://www.nebrija.es/~cmalagon/ia/transparencias/busqueda_heuristica.pdf>.

Berzal, F. (s.f.). Búsqueda heurística. Consultado el 24 de noviembre de 2021. Disponible en <https://elvex.ugr.es/decsai/iaio/slides/A5%20Heuristic%20Search.pdf>

Montero, G. (s.f.). Qué es la heurística: Aprende a aplicar el método Heurístico en el diseño web. Consultado el 24 de noviembre de 2021. Disponible en <https://salyseo.com/diseno-ux-ui/10-leyes-heuristica-user-experience/>

Descubriendo la Inteligencia Artificial (2016). Grafos – Función Heurística. Consultado el 24 de noviembre de 2021. Disponible en <https://www.dropbox.com/s/qaemd4vb4ka3fh9/video.031.funcion.heuristica.pdf?dl=0>

Descubriendo la Inteligencia Artificial (2016). Grafos – Búsqueda Voraz. Consultado el 24 de noviembre de 2021. Disponible en <https://www.dropbox.com/s/y6c84x7c7ekcnz0/video.034.busqueda.voraz.pdf?dl=0>

Descubriendo la Inteligencia Artificial (2016). Grafos – Búsqueda A\*. Consultado el 24 de noviembre de 2021. Disponible en <https://www.dropbox.com/s/kgdq4pbyw0ua007/video.037.busqueda.a.estrella.pdf?dl=0>