Tema 2. Búsqueda en Espacios de Estados

Algoritmos de Búsqueda Heurística: diseño de heurísticos

Objetivos

1. Conocer los fundamentos de los algoritmos de búsqueda y el papel que juegan en la Inteligencia Artificial
2. Conocer el paradigma de Búsqueda en Espacios de Estados y los algoritmos básicos de búsqueda a ciegas y sobre todo de búsqueda inteligente o heurística
3. Saber cómo modelar problemas para resolverlos con Búsqueda en Espacios de Estados, en particular **cómo introducir conocimiento específico del dominio del problema**

Contenidos

2.5. Diseño de heurísticos

- El objetivo es diseñar funciones de la forma

  h: Estados $\rightarrow$ R^+

  que estimen lo mejor posible el valor de h* y que se puedan calcular de forma eficiente (en tiempo polinomial)

- Métodos más comunes
  - Cálculo de cotas inferiores razonando a partir del conocimiento del problema
  - Método de la relajación del problema
  - Otros métodos
    - Aprendizaje automático

2.5.1. Estimación de cotas inferiores razonando a partir del conocimiento del problema

- Ejemplo 1
  - *En el problema del 8-puzzle, un cota inferior del coste de la solución es el número de fichas que no están en la posición que les corresponde en el objetivo.*
  - *Esto es evidente, ya que cada una de estas fichas se tiene que mover al menos una vez.*
- Ejemplo 2
  - *En el TSP, la suma de los costes de los arcos de coste mínimo que incluyen a cada una de las ciudades que tenemos que abandonar también es una cota inferior del coste del problema que representa un estado.*
  - *También es evidente ya que para llegar a una solución tenemos que incluir un arco que permita abandonar a cada una de estas ciudades.*

- Con este método se obtienen heurísticos admisibles, pero tendremos que probar si son o no consistentes, o si se pueden establecer relaciones de dominancia con otros heurísticos para el mismo problema.
- Además, cada problema requiere un razonamiento específico.

2.5.2. El método de la Relajación del Problema
Introducción

- Dos razonamientos para llegar al mismo heurístico h_2 para el problema del 8-puzzle
  - **Razonamiento 1:** *Para llevar una ficha a su posición en el objetivo, ésta tendrá que seguir una trayectoria que, en el caso más favorable, tendrá una longitud igual a la distancia ortogonal de la ficha a su posición en el objetivo.*
  - **Razonamiento 2:** *Si pudiésemos mover las fichas a posiciones que están ocupadas, podríamos llevar cada ficha a su posición en el objetivo, con independencia de las demás, a través de una trayectoria con una longitud igual a la distancia orthogonal.*

- En el primer caso hemos utilizado el método anterior para establecer una cota inferior del coste del problema.
- Y en el segundo lo que hemos hecho es imaginar una versión simplificada del problema y dar un método para calcular una solución óptima del problema simplificado; esta es la base del método de la relajación del problema.

2.5.2. El método de la Relajación del Problema
Descripción y propiedades

- **El método consiste en los siguientes pasos**
  1. Dado un problema ***P*** con un conjunto de restricciones ***R*** *(ojo **R** son restricciones, no reglas),* se eliminan las restricciones de un subconjunto ***R'*** $\subset$ ***R*** de tal forma que se obtenga un problema simplificado, o relajado, ***P'***, que se pueda resolver en tiempo polynomial (clase P).
  2. Se idea un algoritmo que resuelva de forma exacta el problema relajado ***P'*** en tiempo polinomial.
  3. El coste óptimo del problema relajado se utiliza como estimación del coste óptimo del problema real. Este es el valor del heurístico. Es decir:

$$h'(n) = \text{coste de la solución óptima de } n_{P'}$$

$n_{P'}$ es la instancia de ***P´***correspondiente a la relajación de la instancia de ***P*** que representa el estado *n.*

- **Propiedades del método**
  - Los heurísticos son admisibles: esto es claro ya que las soluciones de ***P*** también lo son de ***P'*** pero no al revés.
  - Y también son monótonos: esto no es tan evidente, pero se puede demostrar.
  - Además, si tenenos dos relajaciones ***R''*** $\subset$ ***R'*** $\subset$ ***R*** que dan lugar a los heurísticos *h''* y *h'* respectivamente, se cumple que $h''(n) \geq h'(n)$ para todo estado *n*. Es decir, *h''* domina ampliamente a *h'*.

2.5.2. El método de la Relajación del Problema
Aplicación al problema del 8-puzzle

- **Enunciado del (sub)problema que representa un estado**
  - Se trata de encontrar una secuencia de movimientos de longitud mínima para llegar desde una situación de 8 fichas en un tablero 3×3 hasta una configuración objetivo, de modo que cada movimiento elemental de una ficha desde una casilla A hasta otra casilla B cumpla las 3 restricciones siguientes:

    R_1. Las casillas A y B deben ser adyacentes

    R_2. Las casillas A y B deben estar en la misma fila o en la misma columna, es decir deben ser ortogonales.

    R_3. La casilla B debe estar vacía.

- **Relajaciones posibles**
  - $\{R_1, R_2, R_3\} \Rightarrow h_1$ El problema relajado se resuelve moviendo diretamente cada ficha a su posición en el objetivo
  - $\{R_3\} \Rightarrow h_2$ Cada ficha se lleva al objetivo a través de una trayectoria ortogonal
  - $\{R_1, R_2\} \Rightarrow$ ¿?
  - ¿ . . . ? $\Rightarrow h_3$

2.5.2. El método de la Relajación del Problema
Aplicación al problema TSP

- **Enunciado del (sub)problema que representa un estado**
  - Dado un estado (A{…}X) en el que se han visitado $k \geq 1$ ciudades y estamos en la ciudad X, se trata de calcular un subconjunto de arcos del grafo residual, de coste mínimo, que cumpla las restricciones siguientes:

    R_1. Que tenga N-k+1 arcos.

    R_2. Que los arcos toquen a las ciudades A, X y a las no visitadas .

    R_3. Que el conjunto de arcos sea de grado 1 para A y X, y de grado 2 para las no visitadas.

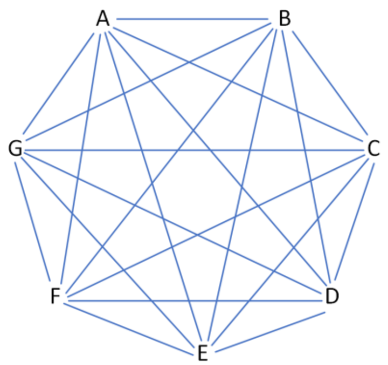
    R_4. Que los arcos conecten a todas las ciudades A, X y las no visitadas entre sí.

  El *grafo residual* se obtiene del grafo de conexiones eliminando las ciudades intermedias del estado y los arcos que tocan a estas ciudades.

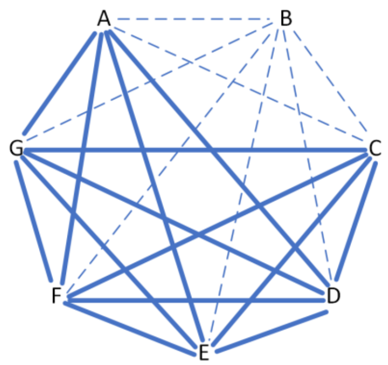
- **Relajaciones posibles**
  - $\{R_1, R_2, R_3, R_4\} \Rightarrow h_0$ La solución del problema relajado es el conjunto vacio.
  - $\{R_2, R_3, R_4\} \Rightarrow h_1$ La solución del problema relajado es el subconjunto de los N-k+1 arcos de coste menor del grafo residual.
  - $\{R_3, R_4\} \Rightarrow h_2$ ¿Existe un algoritmo polinomial que resuelve el problema?
  - $\{R_3\} \Rightarrow h_3$ El problema relajado es un problema bien conocido ¿Qué algoritmo lo resuelve y con qué coste?
  - $\{R_4\} \Rightarrow h_4$ ¿Cómo es el problema relajado y como se resuelve? Este no es tan conocido como el anterior.

2.5.2. El método de la Relajación del Problema

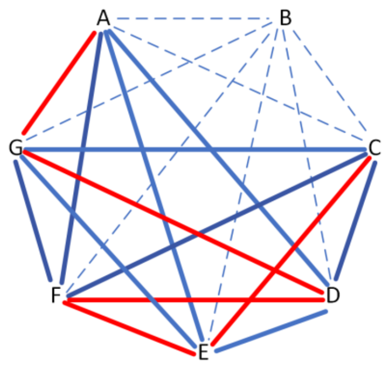
Aplicación al problema TSP (forma de las soluciones del problema real y de los problemas relajados)



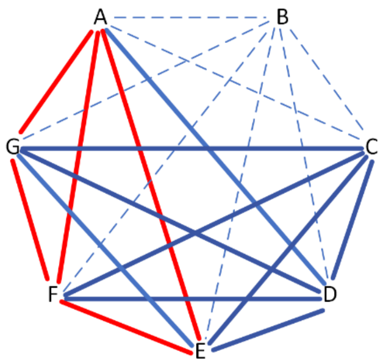
Grafo de conexiones del problema original



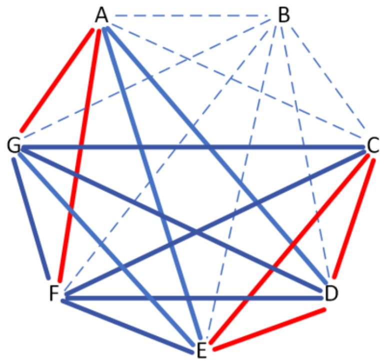
Grafo residual del estado (A {B} C)



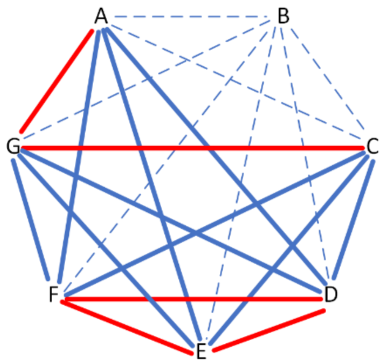
Una solución del problema real (A {B} C)



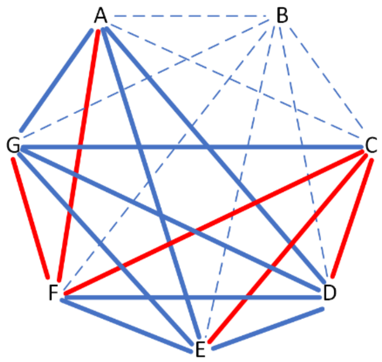
Una solución de la relajación R_2 , R_3 , R_4



Una solución de la relajación R_3 , R_4



Una solución de la relajación R_4



Una solución de la relajación R_3

2.5.3. Aprendizaje de heurísticos

- Dado que un heurístico es una función de la forma

  h: Estados $\rightarrow$ R$^+$

  se pueden utilizar distintos métodos para aprender funciones que aproximen h^*.
- Para ello, el primer paso es disponer de un conjunto de entrenamiento, formado por una serie de pares de la forma

  [estado, *h(estado)]**

  de forma que el **estado** esté caracterizado por una serie de atributos $(at_1,\ldots,at_n)$
- Y luego aplicar algún método de aprendizaje supervisado, como por ejemplo
  - *Regresión*
  - *Redes Neuronales*
  - *Programación Genética*

Aprendizaje de heurísticos. Ejemplo de uso de regresión en el 15-puzzle

[Stern et al. 2014] Potential-based bounded-cost search and Anytime Non-Parametric A. R. Stern, A. Felner, J. Van Den Berg, R. Puzis, R. Shah, K. Goldberg. Artificial Intelligence 214, 1-25. 2014*

- Dado un conjunto de 52,523 pares de valores $[h_2(n), h^*(n)]$ obtenidos a partir de la aplicación de A* a 1,000 instancias, se observa la correlación que existe entre h_2 y h^*. La recta que mejor se ajusta es la que se ve en la figura

R. Stern et al. / Artificial Intelligence 214 (2014) 1–25

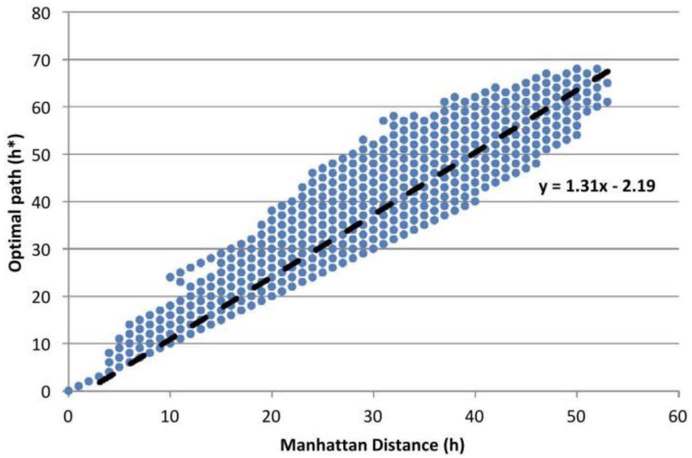


Fig. 2. MD heuristic vs. true distance for the 15-puzzle domain.

- A partir de este resultado, podemos definir un nuevo heurístico como:

$$h(n) = max(0,\ 1.31 \times h_2(n) - 2.19)$$