



Diseño de Algoritmos

Sesión 15

Profesores:

Tomás Lara Valdovinos – t.lara@uandresbello.edu

Jessica Meza-Jaque – je.meza@uandresbello.edu

OBJETIVOS DE LA SESIÓN

- Conocer y aplicar conceptos de heurística
- Conocer el algoritmo A*



CONTENIDOS DE LA SESIÓN



- Importancia de la heurística para resolver problemas computacionales
- Algoritmo A*

RECORDEMOS

¿Qué es un Algoritmo?

Es un
PROCEDIMIENTO

DEFINIDO (bien
especificado)
ORDENADO, y
FINITO

Para RESOLVER UN
PROBLEMA,
transformando una
instancia de ENTRADA
en una SALIDA

Enfoquémonos en la siguiente parte de la definición

DEFINIDO (bien
especificado)

ORDENADO, y

FINITO

¿Qué sucede cuando la descripción y solución de un problema no podemos encasillarla dentro de esas características?

¿Qué ejemplos podemos destacar?

Por ejemplo...

- ¿Cómo tomo una decisión algorítmica cuando no conozco que podría ocurrir en el futuro?
 - ¿Cuánto tiempo me tomará llegar a mi destino?
 - ¿Qué obstáculos habrán en el camino?
 - ¿Llegaré?

Veamos el siguiente video

- <https://www.youtube.com/watch?v=pJI8LfnPPB8>

Autoría: *Martin Schleicher.*

Heurística

- Es una manera de resolver un problema utilizando experiencias u otras estrategias llevadas a lógica, para concretarlas de una manera más corta y a veces aproximada.

Ejemplo

- Un jugador de ajedrez utiliza un pensamiento heurístico antes de realizar una jugada. Por ejemplo:
 - Salvar a la reina amenazada.
 - Elegir entre sacrificar un peón o un alfil.
 - Usar una jugada que en juegos anteriores lo acercaron más a una victoria.

Influencias (Principio de representatividad)

- Las personas resolvemos problemas tomando decisiones heurísticas más que algorítmicas.
 - ¿Compro la bebida que vi en la televisión o la más barata?
 - ¿Entro al restaurant más lleno o al más vacío?

"cuando las personas tratamos de calcular la probabilidad de un evento, lo hacemos en función de la representatividad, es decir, la proximidad de este evento con otro que es típico"

Puente Ferreras, 1998.

¿Cómo resuelvo un problema utilizando heurísticas?

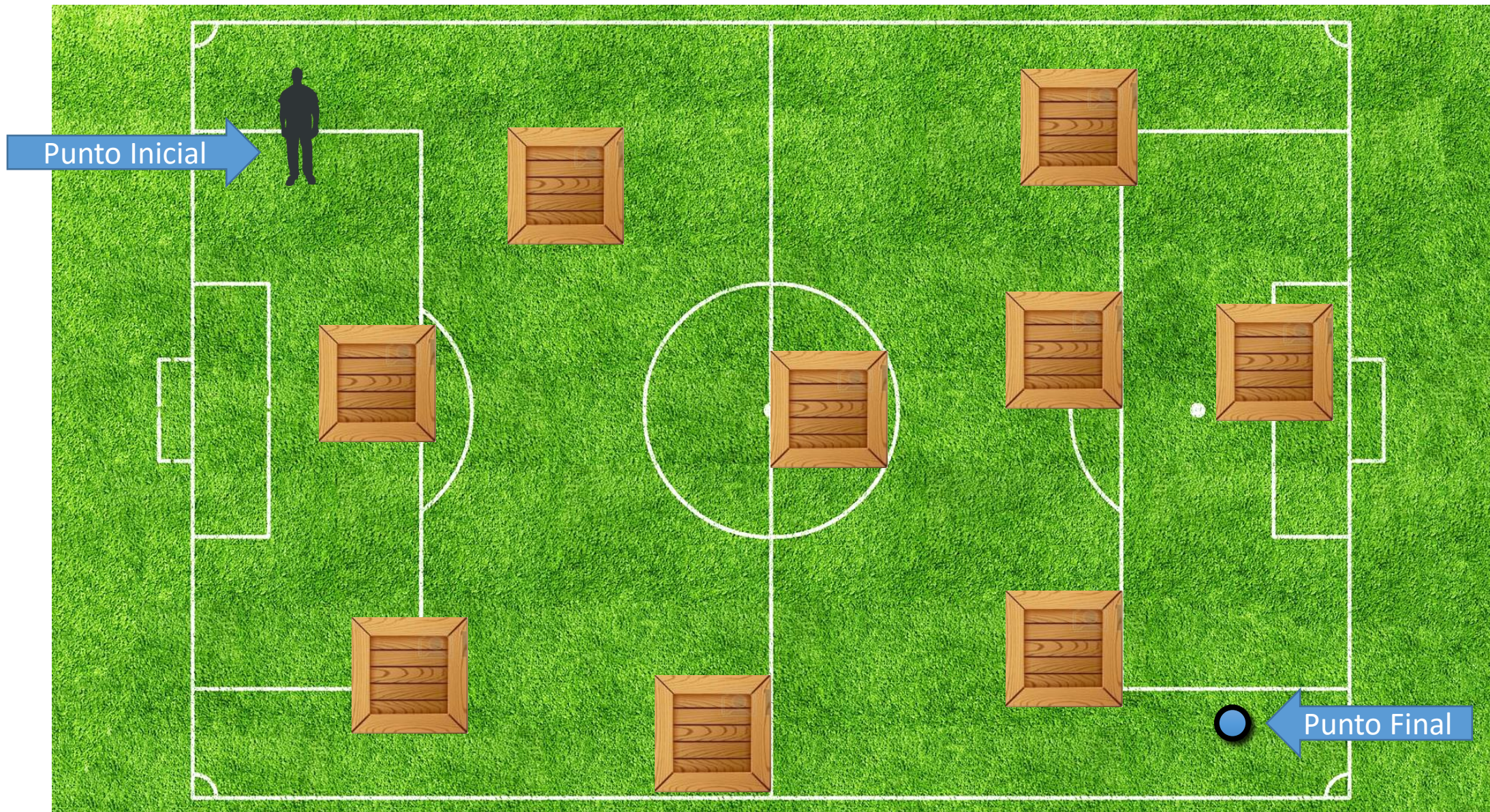
Usando:

- Imaginación
- Memoria
- Inteligencia
- Tentativas
- Improvisación
- Flexibilidad
- Criterio
- Toma de decisiones en contexto de incertidumbre
- Tolerancia a los errores
- Capacidad para reponernos del fracaso



Aplicación de la heurística en diseño de algoritmos

Supongamos el siguiente caso visto anteriormente



¿Cómo resolverías el problema usando una estrategia heurística?

Algoritmo A*

- Pronunciado A-Star (A estrella) o A-Asterisco.
- Encuentra la ruta de menor coste siempre que se cumplan ciertas condiciones.
- Utiliza la heurística para predecir la mejor ruta a su destino.



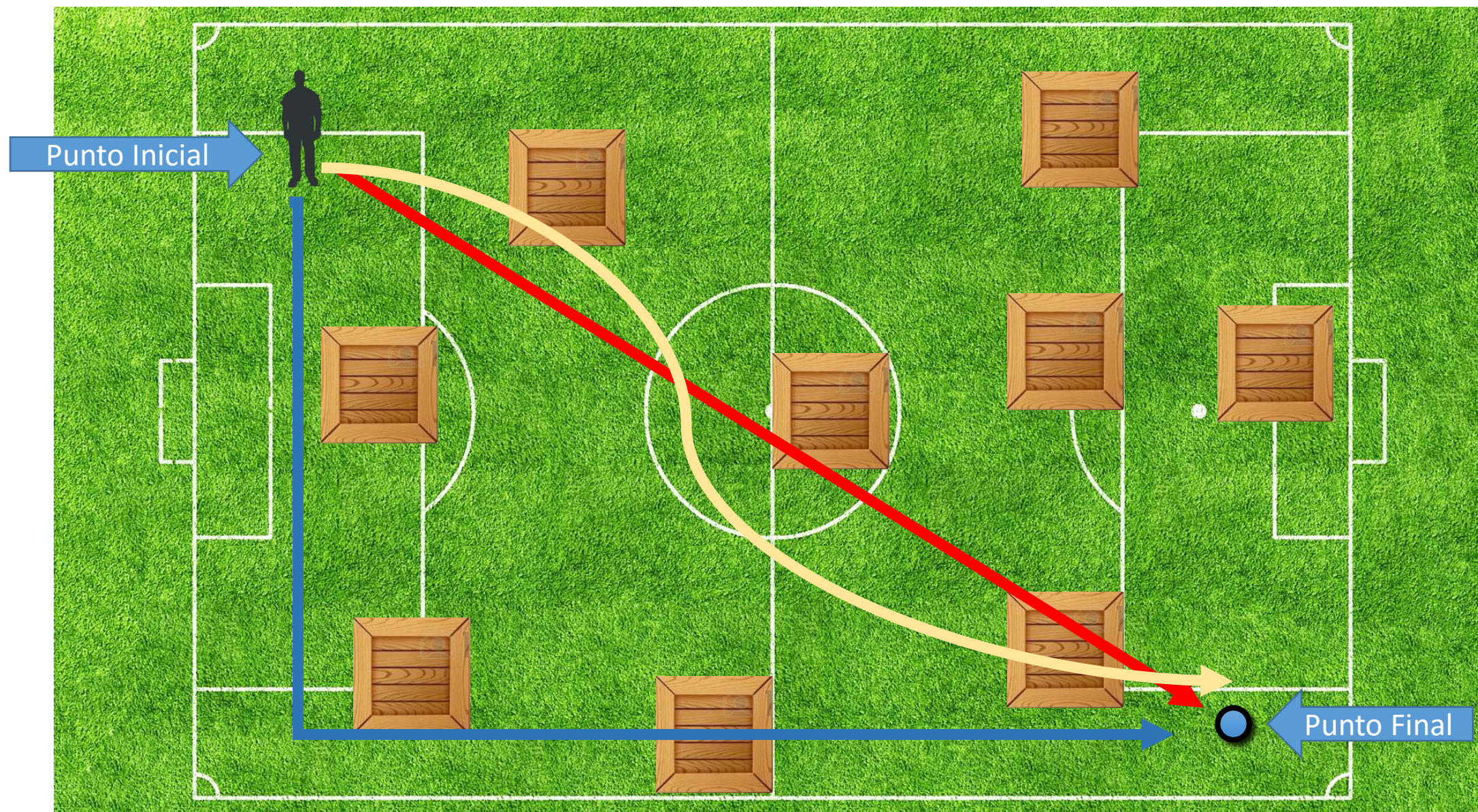
“Por convención, los algoritmos que en su nombre contengan un asterisco utilizan heurísticas.”

Heurística admisible

- A* utiliza la heurística admisible.
- Se dice que una heurística es admisible si nunca sobreestima el costo de alcanzar el objetivo, o sea, que en el punto actual la estimación del costo de alcanzar el objetivo nunca es mayor que el menor costo posible.

Extraído de Wikipedia

¿Cuál es o sería la mejor ruta para alcanzar el objetivo en todo momento?



Heurística en A*

- Seguir en una línea recta hacia el objetivo podría ser la mejor ruta si no nos topamos nunca con un obstáculo.
- También podríamos zigzaguear y disminuir la probabilidad de colisionar en la mayoría de los casos.
- Podríamos decidir doblar en una esquina basado en un aprendizaje previo del entorno.

Heurística en A*

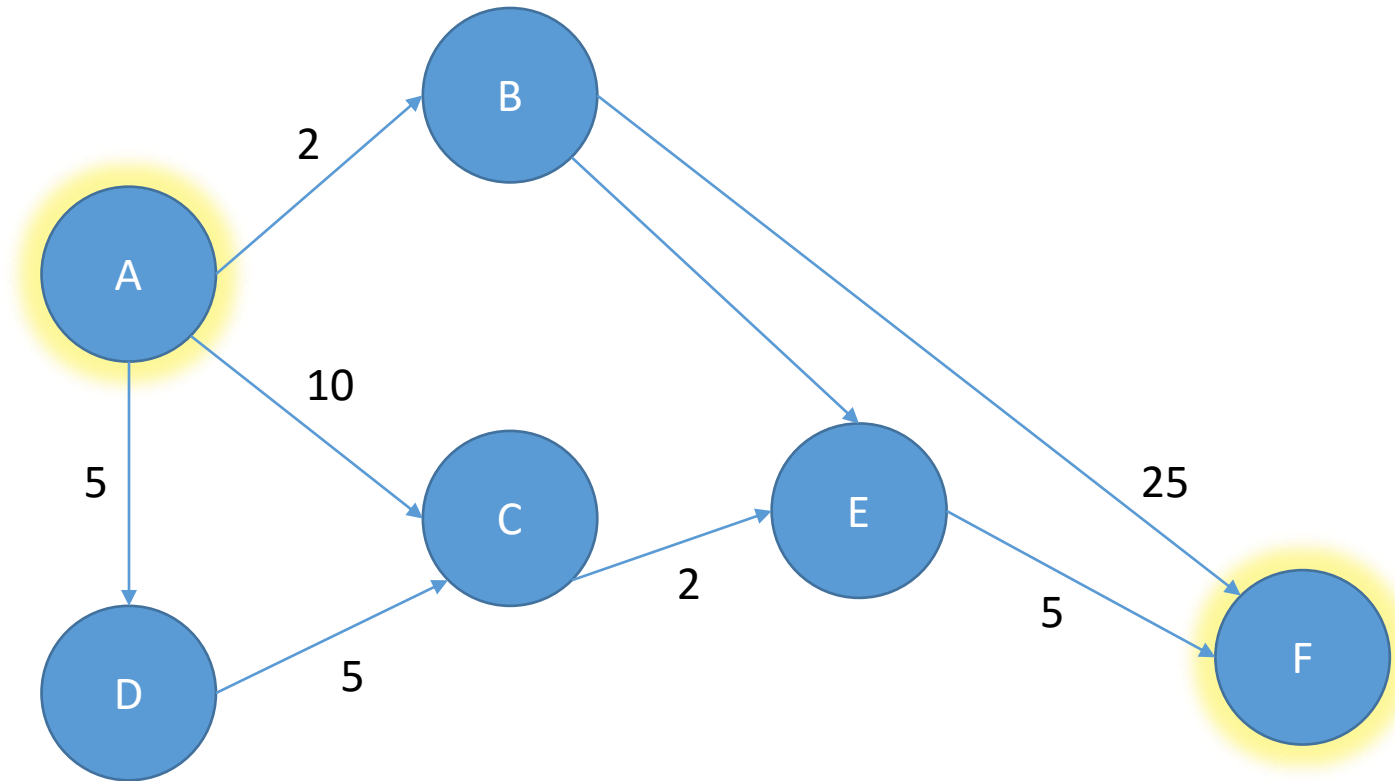
- A* utiliza en una función evaluativa que indica 2 factores, ésta es:
- $F^*(x) = G^*(x) + H^*(x)$

$G^(x)$ estima la secuencia de movimientos más corta desde el inicio hasta x , basándose en lo recorrido hasta el momento.*

$H^(x)$ estima la secuencia de movimientos más corta desde x al punto final, basándose en una heurística.*

$F^(x)$ estima la secuencia de movimientos más corta desde el inicio al punto final pasando a través del punto x .*

Ejemplo



Queremos llegar desde el vértice **A** al vértice **F**

¿Desde qué punto comenzamos a evaluar?

Dijkstra

- Si tomamos en consideración como lo hacía Dijkstra tendríamos que:
 1. Tomar todos los puntos vecinos del nodo inicial y ponerlos en una cola de prioridad.
 2. El vecino más cercano (el que tiene el menor coste) es el primero en ser evaluado, en este caso el nodo B.
 3. Luego seguimos con el nodo cuyo sumatoria del coste desde el punto de origen sea menor en la cola de prioridades.
 4. Y así evaluamos hasta llegar al nodo destino.

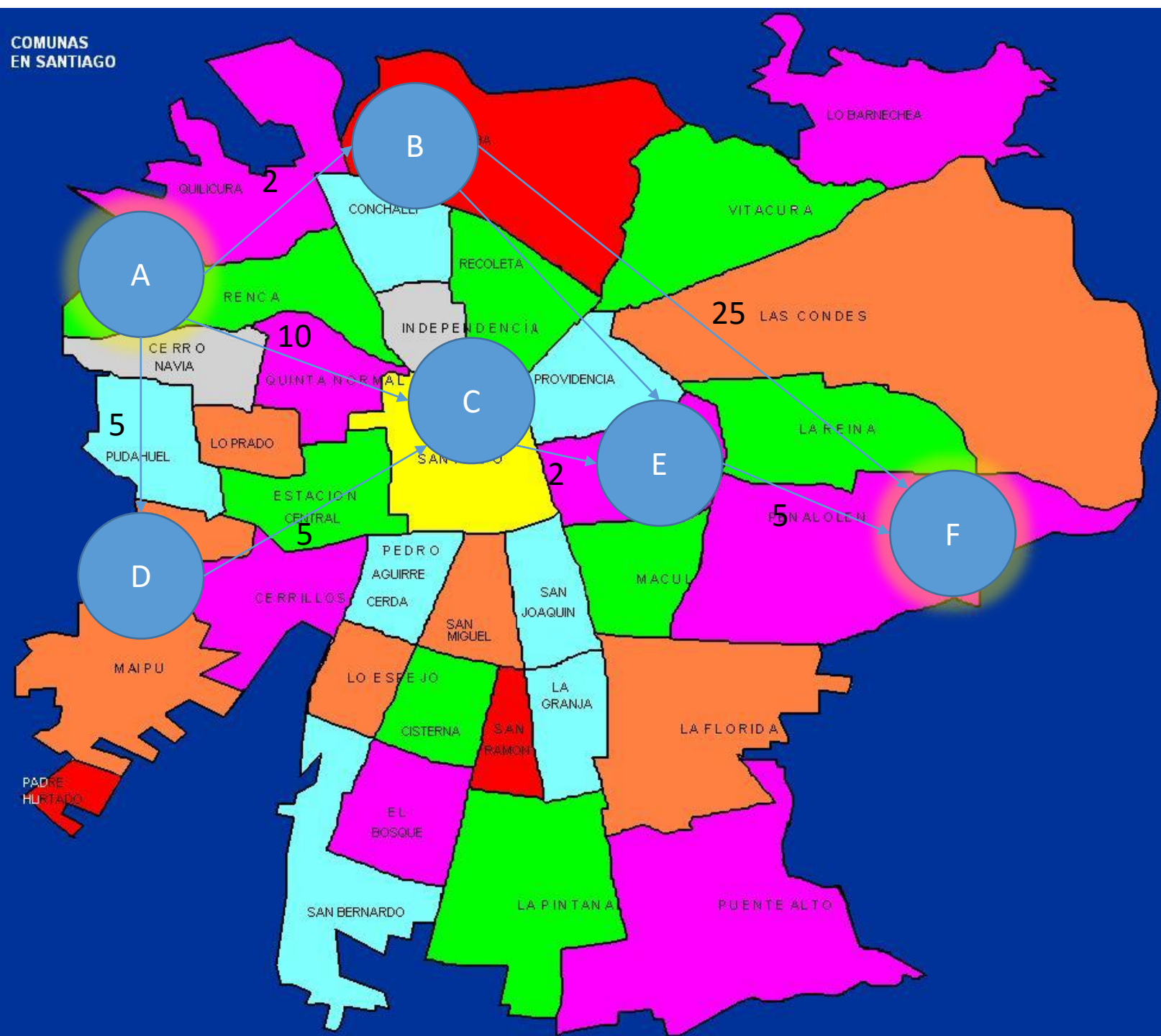
¿Qué sucede con Dijkstra?

¿Qué sucede con Dijkstra?

- Por lo general, este algoritmo requiere evaluar una gran cantidad de caminos antes de llegar al nodo final, porque está basado en las distancias recorridas desde el punto de inicio, sin utilizar ninguna información acerca de la ruta posible a seguir hasta el nodo final.

¿Cómo cambia nuestra percepción del problema si ahora vemos esto?

COMUNAS
EN SANTIAGO



Medidas geográficas

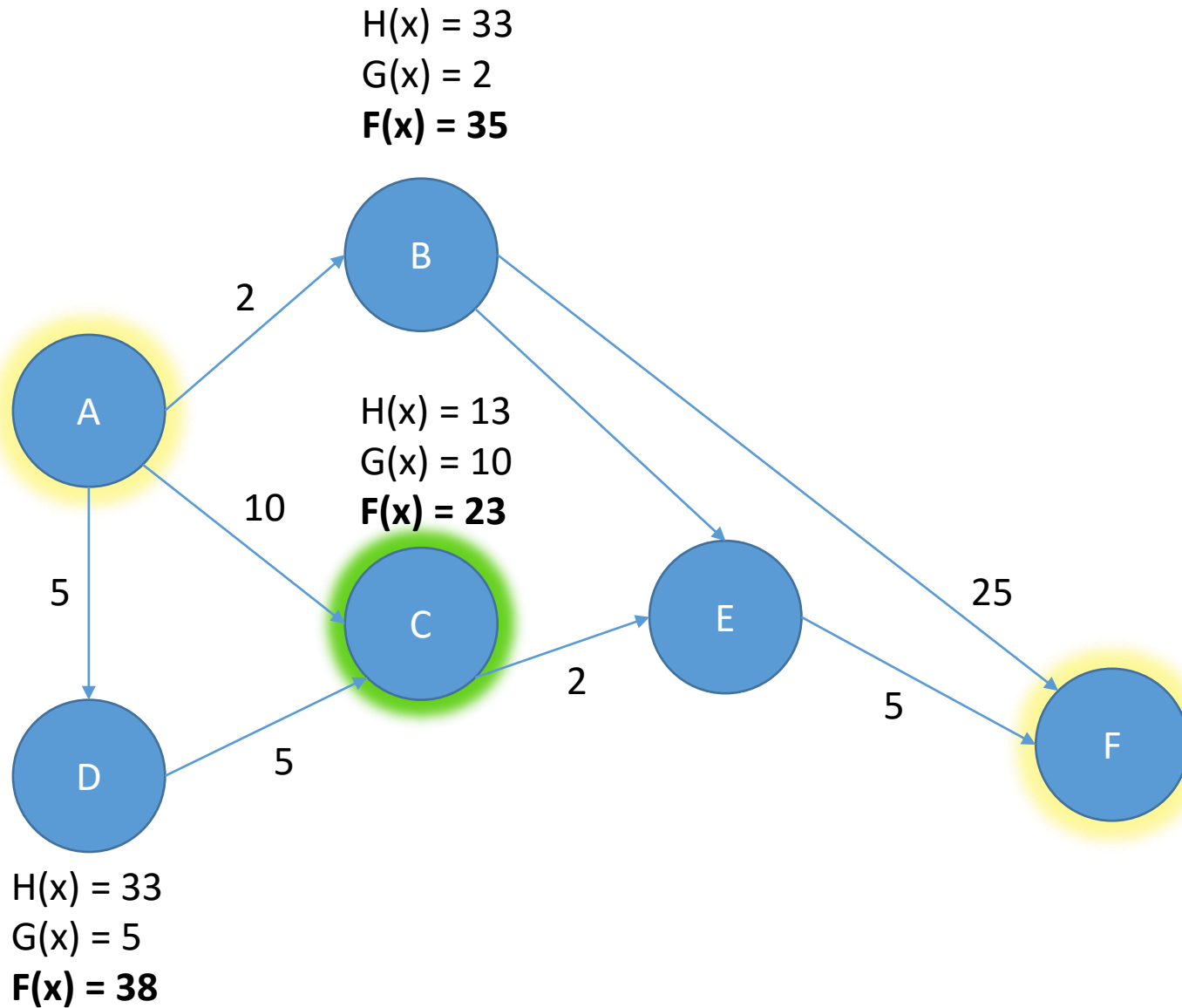
- Si vemos el mapa, nuestro vértice destino se encuentra en la comuna de Peñalolén.
- Mientras que los vecinos del punto inicial (B, C y D) están en las comunas de Huechuraba, Santiago Centro y Maipú respectivamente.
- Si tomáramos una regla y midiéramos las distancias de estas 3 comunas con respecto a la comuna final tendríamos que:
 1. Huechuraba -> Peñalolén \sim 33 Kms.
 2. Santiago Centro -> Peñalolén \sim 13 Kms.
 3. Maipú -> Peñalolén \sim 33 Kms.

Medidas geográficas

- Por lo tanto podríamos decir que hay más posibilidades de evaluar Peñalolén antes si evaluamos primero Santiago Centro (Nodo C).
- Esto es independiente de los nodos que resten por visitar si tomáramos la decisión de ir por C (por ejemplo, el nodo E no fue tomado como factor para llegar a Peñalolén).

Heurística

- En este caso, esa suposición de “por dónde” podríamos llegar antes al punto final es lo que conocemos como heurística en el algoritmo A*.
- Este es el valor que denotamos por $H^*(x)$



Elección del camino más próximo

Para decidir qué nodo evaluaremos en la siguiente iteración desde A tomaremos como $H(x)$ al valor heurístico entre ese nodo y el final.

El valor de $G(x)$ será el coste que tenemos de llegar a ese nodo desde el punto inicial (Nodo A), igual como lo hacíamos con Dijkstra.

El valor de $F(x)$ será igual a la suma entre G y H .

El nodo que tenga una valor de F menor será el que tendrá mayor prioridad para ser evaluado.

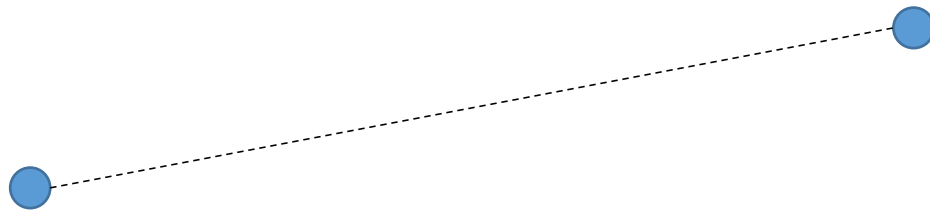
En este caso, el nodo C es el que posee el menor valor de F , por lo tanto seguimos evaluando desde ahí.

Técnicas heurísticas en A^*

- Distancia Euclidiana
- Distancia Manhattan

Distancia Euclidiana

- Se basa en que la mejor distancia entre un punto y otro es una línea recta.
- Es útil cuando no tenemos más opciones y la cantidad de obstáculos entre los puntos es mínima.



Distancia Manhattan

- Se basa en la ciudad de Manhattan, la cual tiene sus cuadras perfectas y para llegar de un punto a otro lo podemos hacer de manera tal de seguir derecho y doblar en una sola esquina.



Otras medidas heurísticas

- Podemos usar otros datos como:
 - Cantidad de autopistas en el recorrido.
 - Promedios de congestión
 - Horario del día.
 - Caminos que por experiencia sabemos que son mejores.
 - Siempre he tomado esa ruta y ningún robot me va a decir por donde debo irme.

Algoritmo en Pseudocódigo

OPEN //Cola de prioridades de nodos a evaluar, el valor de $F(x)$ es su prioridad

CLOSED //Los nodos ya evaluados

add **S** to **OPEN**

loop

current = node in **OPEN** with the lowest **f_cost**

 remove **current** from **OPEN**

 add **current** to **CLOSED**

 if **current** is the **target** node //path has been found

 return

 foreach **neighbour** of the **current** node

 if **neighbour** is in **CLOSED**

 skip to the next **neighbour**

 if new path to **neighbour** is shorter OR **neighbour** is not in **OPEN**

 set **f_cost** of **neighbour**

 set **parent** of **neighbour** to **current**

 if **neighbour** is not in **OPEN**

 add **neighbour** to **OPEN**

Demostración y comparación con Dijkstra

- <https://www.youtube.com/watch?v=cSxnOm5aceA>

CHECK - OBJETIVOS DE LA SESIÓN

- Conocer y aplicar conceptos de heurística
- Conocer el algoritmo A*

CHECK





Diseño de Algoritmos

Sesión 15

Profesores:

Tomás Lara Valdovinos – t.lara@uandresbello.edu

Jessica Meza-Jaque – je.meza@uandresbello.edu