



INTELIGENCIA ARTIFICIAL

Facultad de Ingeniería de Sistemas
e Informática (FISI)

Universidad Nacional Mayor de San Marcos
(UNMSM)

Agenda

- ❖ Búsqueda informada
 - ✓ Primero el Mejor (avara, costo uniforme, A-estrella)
- ❖ Ramificación y Acotación
- ❖ Ascenso a la Colina
- ❖ Resolución de Problemas

Familia de métodos Primero el Mejor

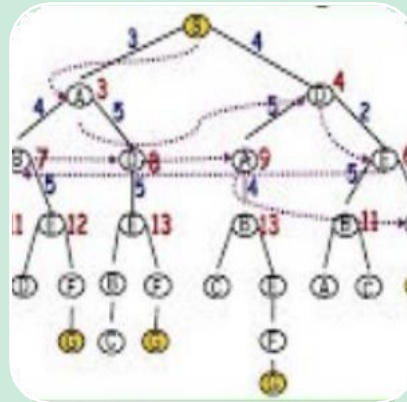


Busq. Avara

$$f(n) = h(n)$$

$h(n)$: costo estimado de la ruta desde el nodo n al nodo meta

$h(n) = 0$, si n es el nodo meta.

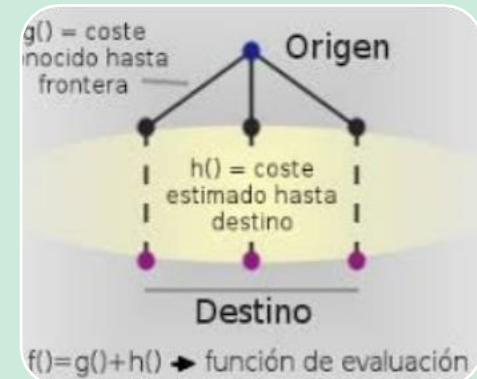


Busq. Costo Uniforme

$$f(n) = g(n)$$

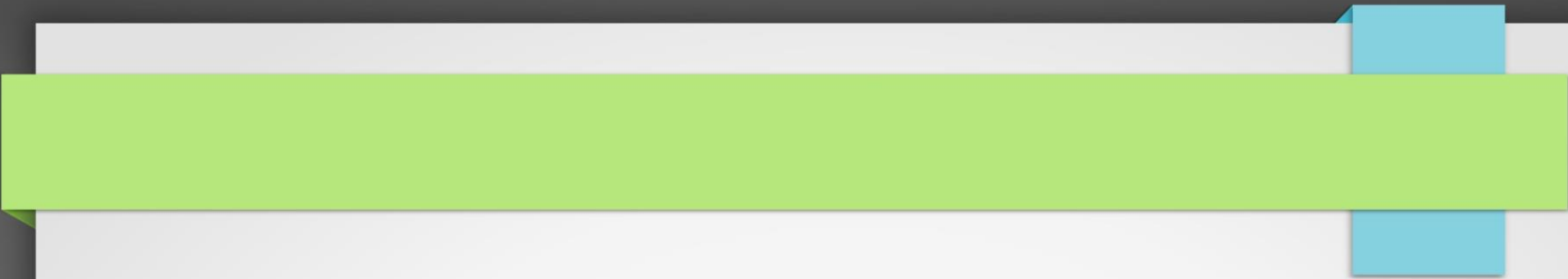
$g(n)$: costo desde el nodo raíz hasta el nodo n .

$g(n)$ para algunos problemas, es igual al nivel del árbol al que pertenece el nodo n



Busq. A-estrella

$$f(n) = g(n) + h(n)$$



Método de Ramificación y Acotación

Conceptos: Ramificar y Acotar

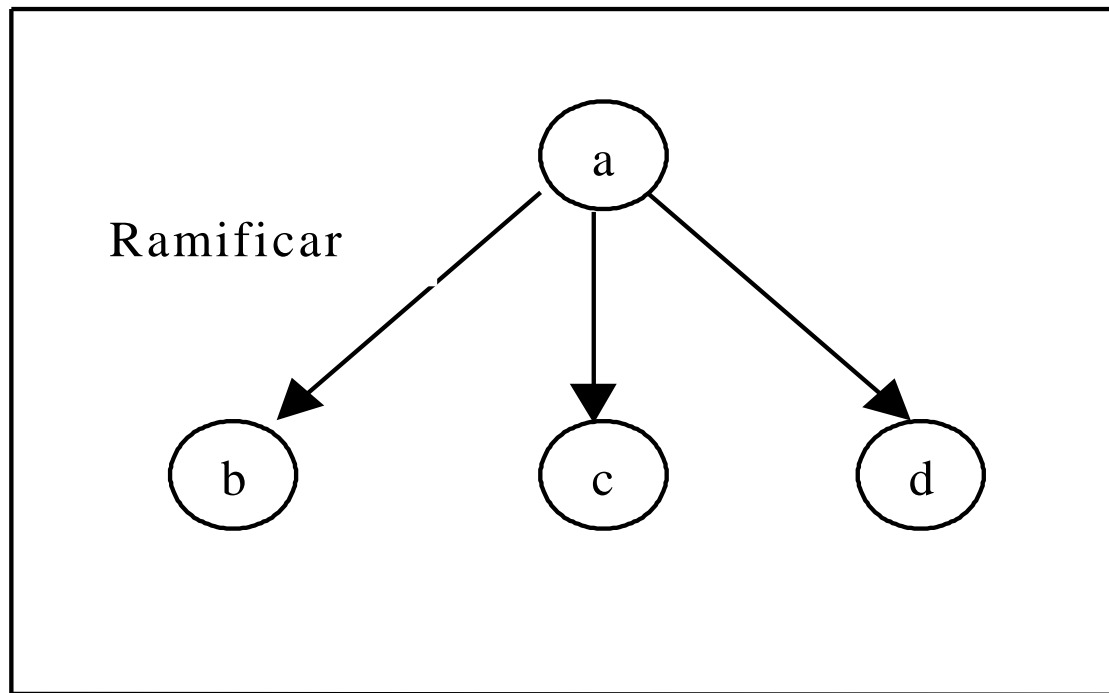
Ramificar

Se entiende por **ramificar** un nodo de un árbol al proceso de generar los nodos sucesores a este.

Todos los métodos ciegos y los informados de búsqueda son métodos de ramificación pues todos ellos incluyen un proceso de ramificación

Conceptos: Ramificar y Acotar

Ejemplo - Ramificar



Ramificación del nodo a

Conceptos: Ramificar y Acotar

Criterios de Ramificación

Primero el Mejor

El nodo que presenta “mejor” (mayor o menor) valor de la función de mérito será el primero a ser procesado.

FIFO (First Input First Output)

El primer nodo a entrar en LE será el primero a ser procesado

LIFO (Last Input First Output)

El último nodo a entrar en LE será el primero a ser procesado.

Conveniencia: los nodos se registran al inicio

Conceptos: Ramificar y Acotar

Ejemplo - Criterio de Ramificación

LE	4	5	2	9	9	8
	ABC	ABD	ABEF	EFAG	EDAC	CADFG

Nodo a Ramificar: EFAG o EDAC (mejor = mayor)

Primero el Mejor: ABEF (mejor = menor)

LIFO: ABC

FIFO: CADFG

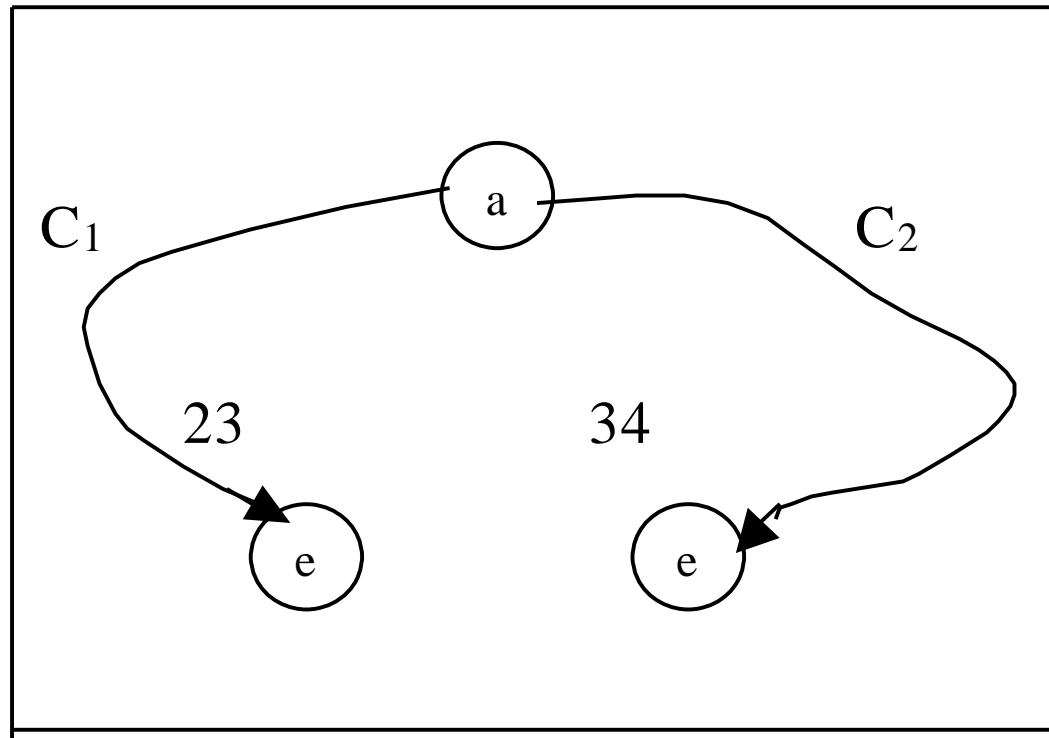
Conceptos: Ramificar y Acotar

Acotación

El proceso de simplificación de la búsqueda a través de la poda de ramas o subárboles del árbol de estado que presentan peores soluciones es llamado proceso de **acotación**.

Conceptos: Ramificar y Acotar

Ejemplo de Acotación



Dos caminos conducen a un mismo nodo

Conceptos: Ramificar y Acotar

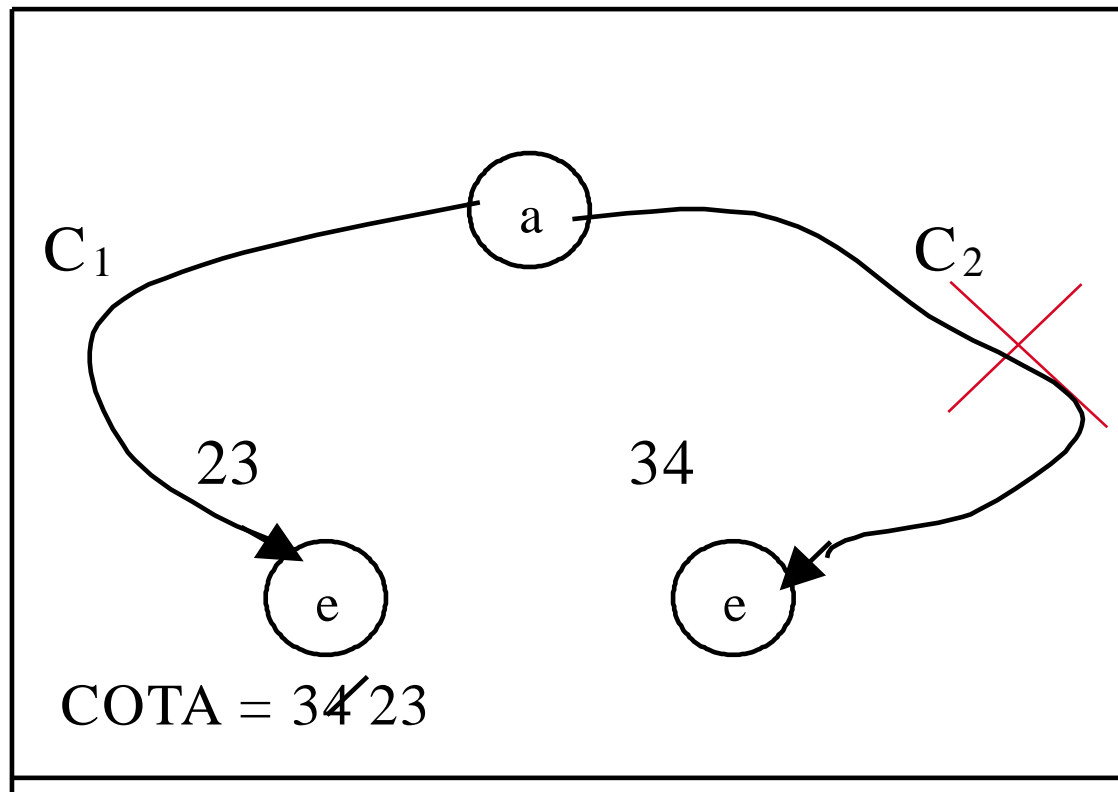
Actualización de la Cota

Asociado a cada estado procesado se tiene un valor de la función de evaluación, a este número se le llama **cota del nodo**.

Si en el proceso de ramificación es generado un nodo ya procesado y con mejor valor de función de evaluación que la cota asociada a este, entonces la cota de este nodo deberá ser actualizada por el valor de la función de evaluación.

Conceptos: Ramificar y Acotar

Ejemplo de Acotación

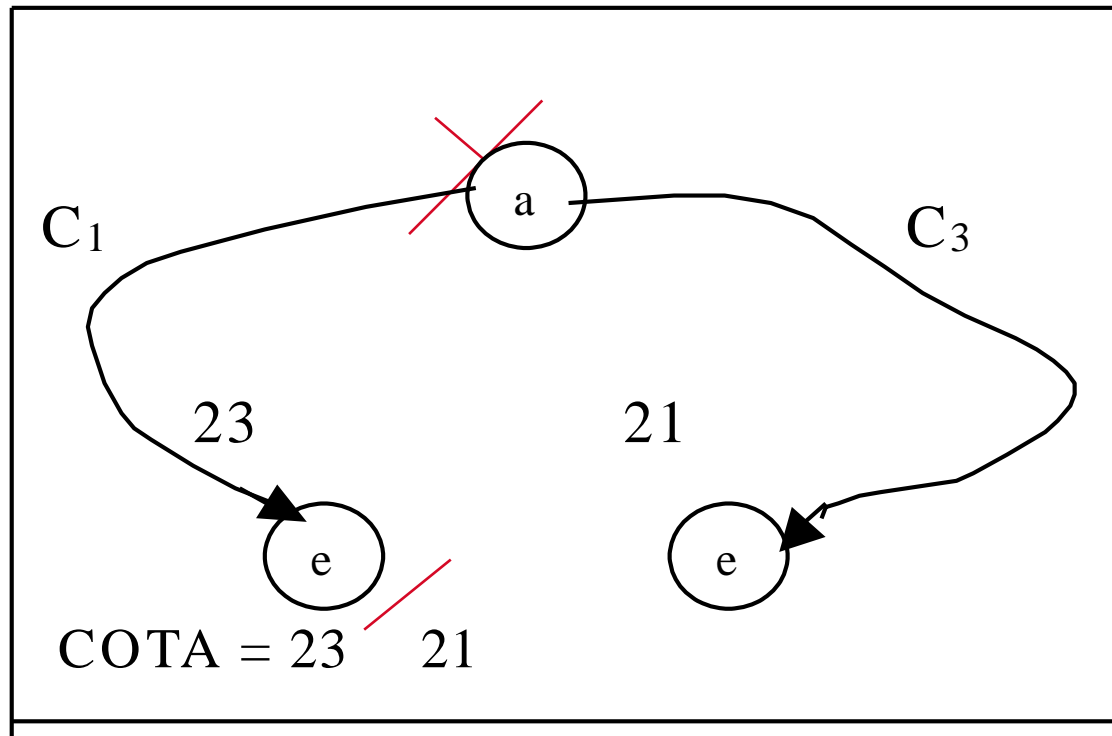


Se escoge el camino más corto para llegar a ese nodo; esto implica actualizar la cota del nodo

Dos caminos conducen a un mismo nodo

Conceptos: Ramificar y Acotar

Ejemplo de Acotación



En este caso se encontró un camino más corto para llegar a ese mismo nodo; esto implica actualizar, otra vez, la cota del nodo

Dos caminos conducen a un mismo nodo

Proced. de Ramificación y Acotación

Implementación – Listas

LE:

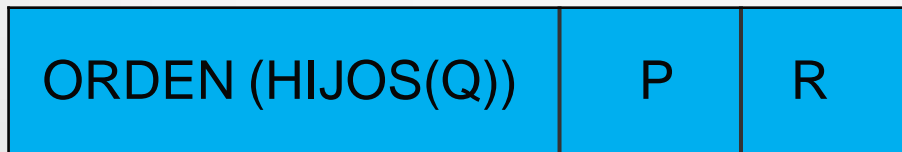
PROCESAR



Criterio de
Ramificación

LE:

REGISTRO



Conveniencia

ACTUALIZAR COTAS: LV

Proced. de Ramificación y Acotación

Algoritmo de Ramificación y Acotación – Primero el Mejor

Inicio

1. LE := ((Estado_Inicial)); LV := (Estado_Inicial); Cota(Estado_Inicial): =0;

Test de Parada:

2. Si (LE = ()) entonces Escribir("no hay solución"), PARE;

3. LISTA := Primero(LE); P := Ultimo(LISTA);

4. Si (P es Meta) entonces Escribir ("solución =", LISTA), PARE;

Ramifica y Acota:

6. Elimina_Mejor(LE);

7. Para (Nodo ∈ Hijos(P))

7.1 W_LISTA=LISTA

7.2 Si (Nodo ∉ LV)

7.2.1 Adiciona_final(Nodo,W_LISTA)

7.2.2 Adiciona_inicio(W_LISTA, LE)

7.2.3 Cota(nodo) := f (W_LISTA)

7.2.4 Adiciona_Ultimo(Nodo,LV)

7.3 Sino

7.3.1 Si (f (W_LISTA) < Cota(Nodo))

7.3.2 Adiciona_final(Nodo,W_LISTA)

7.3.3 Adiciona_inicio(W_LISTA, LE)

7.3.4 Cota(Nodo) = f (W_LISTA)

Fin_Si;

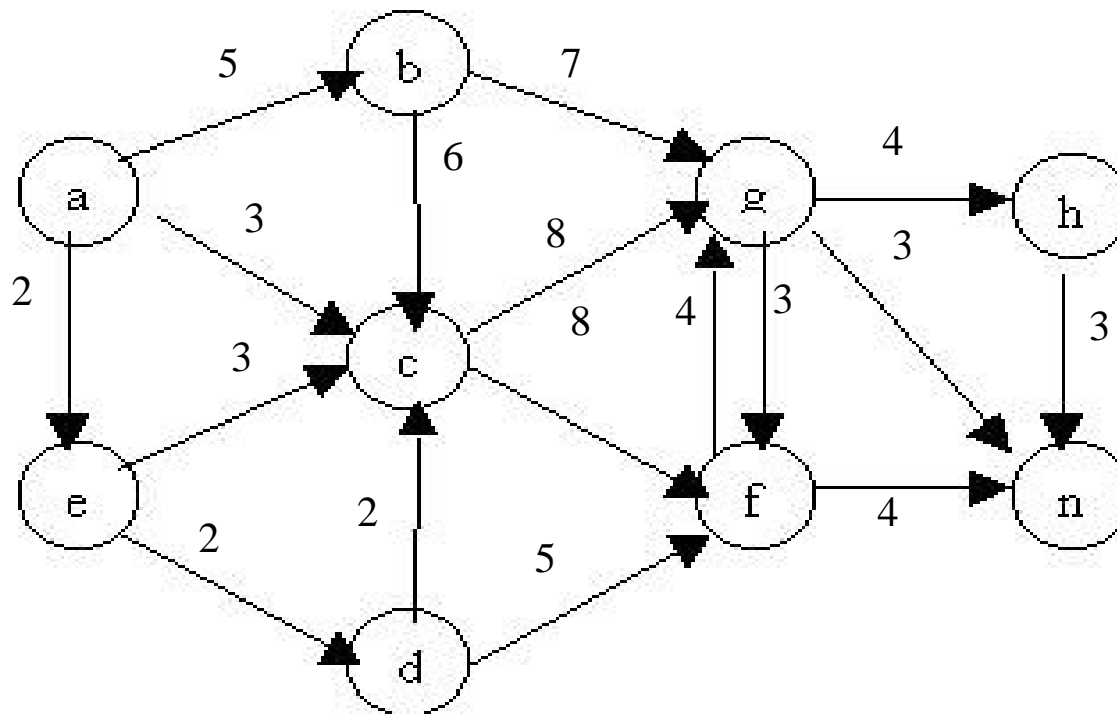
Fin_Para;

8. Ordena(LE)

9. Ir Para 2

Proced. de Ramificación y Acotación

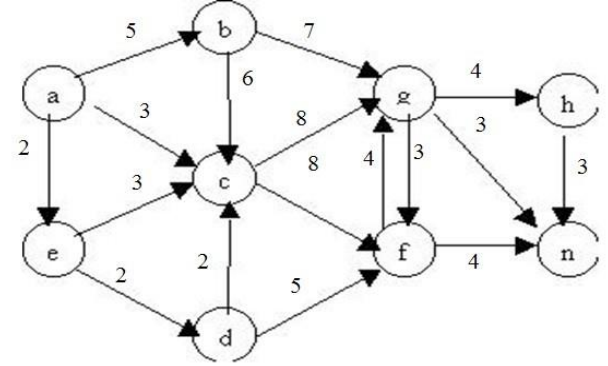
Algoritmo de Ramificación y Acotación – Primero el Mejor



Determine una ruta "a – n" de menor distancia para la red de transporte mostrada

Proced. de Ramificación y Acotación

Fc de evaluación = distancia recorrida



Its	LE	P	LV
1	((a)-0)	a-0	a-0
2	((a e)-2 (a c)-3 (a b)-5)	e-2	a-0 e-2 c-3 b-5
3	((a c)-3 (a e d)-4 (a b)-5))	c-3	a-0 e-2 c-3 b-5 d-4
4	((a e d)-4 (a b)-5 (a c f)-11 (a c g)-11)	d-4	a-0 e-2 c-3 b-5 d-4 f-11 g-11
5	((a b)-5 (a e d f)-9 (a c f)-11 (a c g)-11)	b-5	a-0 e-2 c-3 b-5 d-4 f-9 g-11
6	((a e d f)-9 (a c f)-11 (a c g)-11)	f-9	a-0 e-2 c-3 b-5 d-4 f-9 g-11
7	((a c f)-11 (a c g)-11 (a e d f n)-13)	f-11	a-0 e-2 c-3 b-5 d-4 f-9 g-11 n-13
8	((a c g)-11 (a e d f n)-13)	g-11	a-0 e-2 c-3 b-5 d-4 f-9 g-11 n-13
9	((a e d f n)-13 (a c g h)-15)	n-13	a-0 e-2 c-3 b-5 d-4 f-9 g-11 n-13 h-15

Estado Meta

La ruta solución es a-e-d-f-n con un valor de 13

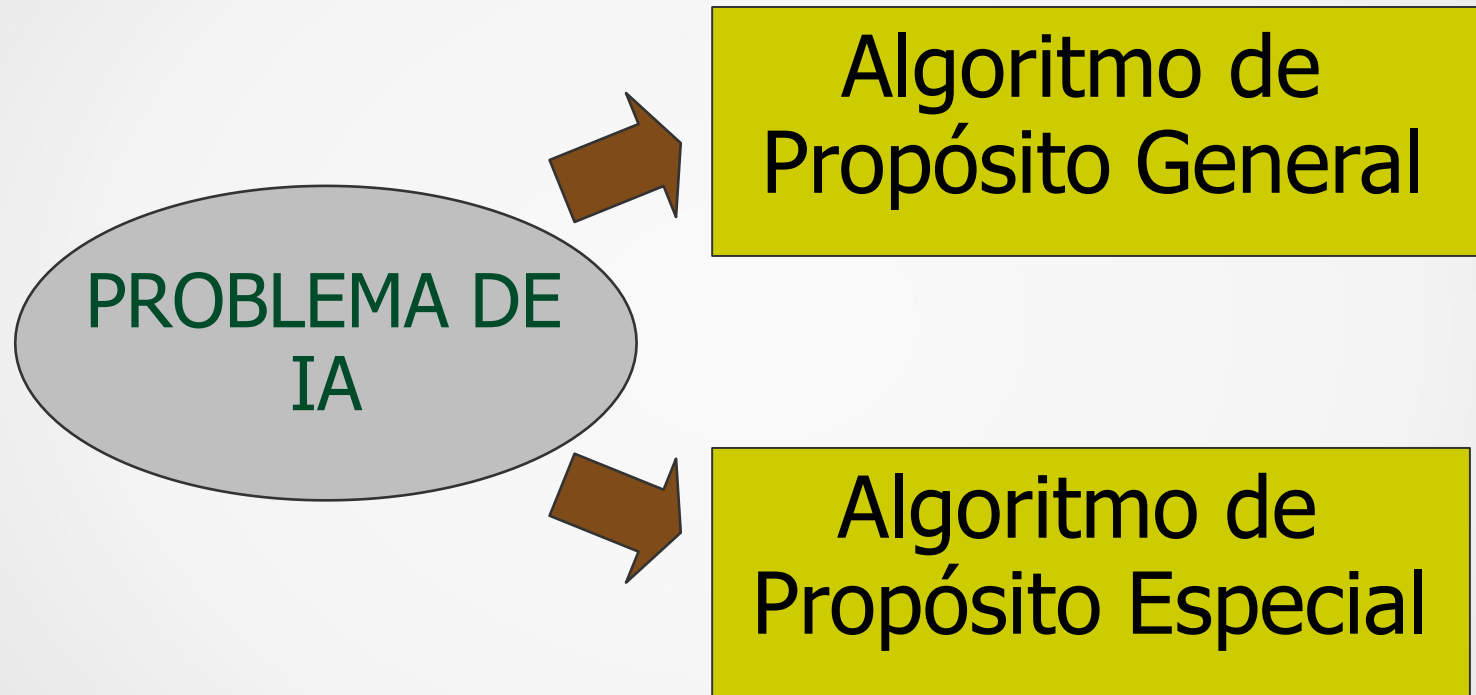
Proced. de Ramificación y Acotación

Observaciones

El método garantiza una solución exacta para problemas de optimización

Este es el método más eficiente de los métodos ciegos y de aquellos que usan información adicional

Resolución de problemas



Resolución de problemas

Algoritmo de Propósito General

- ❖ Se refiere a un algoritmo que pueda servir para resolver cualquier problema de IA.
- ❖ Para problemas de Optimización, en el test de parada se deberá incluir alguna condición de optimalidad
- ❖ Este algoritmo en principio podrá resolver cualquier problema de optimización, sin embargo será por lo general bastante ineficiente

Resolución de problemas

Algoritmo de Propósito General

Parámetros de Entrada

Estado Inicial

Condición de Optimalidad o Test de Parada

Hijos o sucesores

Función de evaluación

Resolución de problemas

Algoritmo de Propósito Especial

- ❖ Se refiere a un algoritmo desarrollado para resolver un problema específico de IA no sirviendo para resolver algún otro problema.
- ❖ Para problemas de optimización en el test de parada se deberá incluir alguna condición de optimalidad, y todos los otros pasos deberán ser modificados aprovechando la estructura particular y específica del problema a resolver.

Resolución de problemas

Algoritmo de Propósito Especial

Este algoritmo presenta una buena eficiencia respecto a los de propósito general, razón por la cual son los más usados para resolver problemas del tipo de optimización.

Resolución de problemas

Algoritmo de Propósito Especial

Parámetros de Entrada

Estado Inicial

Condición de Optimalidad o Test de Parada

Hijos o sucesores

Función de evaluación

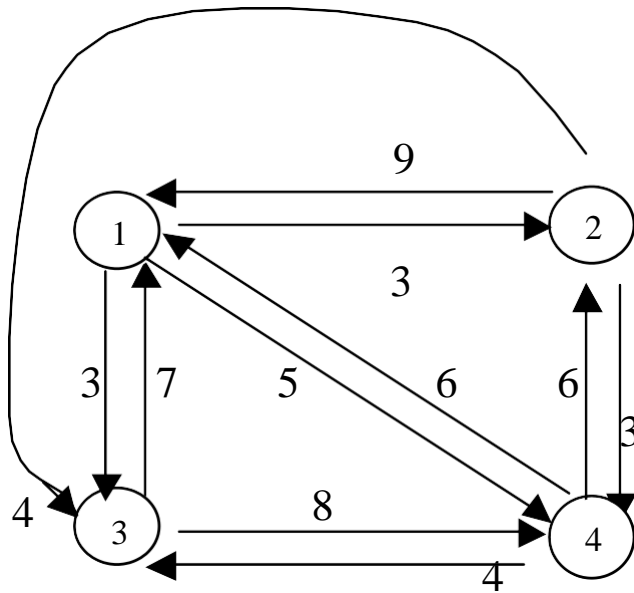
Resolución de problemas

Agente Viajero

Un agente viajero debe recorrer n ciudades una vez por cada ciudad. El agente deberá partir y retornar desde una de las n ciudades. El problema consiste en determinar una ruta de menor suma de distancias para el agente. Suponga conocida las distancias entre las ciudades.

Resolución de problemas

Agente Viajero - Representación



Red de transporte entre ciudades

Ciudad de Origen	Ciudad de Destino			
	1	2	3	4
1	-	3	3	5
2	9	-	4	3
3	7		-	8
4	6	6	4	-

Resolución de problemas

Agente Viajero

Estado

Lista de ciudades que se visita

Estado Inicial

Ciudad de partida
[1]

Resolución de problemas

Agente Viajero

Condición de optimalidad

$L(.)$: nro de ciudades visitadas de una secuencia

LISTA: una secuencia de ciudades que presenta menor suma de distancias

Si $L(\text{LISTA}) = n+1$ entonces LISTA
es una
solución óptima al problema

Resolución de problemas

Agente Viajero

Generación de sucesores

Sea $P = \text{Ultimo}(\text{LISTA})$

$\text{Hijos}(P)$ es el cjto de nodos
sucesores

Si $L(\text{LISTA}) < n$ entonces

$\text{Hijos}(P) = \{1, 2, \dots, n\} - \text{LISTA}$

Si $L(\text{LISTA}) = n$ entonces

$\text{Hijos}(P) = \{1\}$

Resolución de problemas

Agente Viajero

Función de evaluación

Como se desea la ruta de menor distancia: más adecuada es la suma de distancias asociada a una ruta.

Resolución de problemas

Agente Viajero

Acotación

El algoritmo de RyA considera una cota para cada nodo. Para nuestro caso no podemos considerar una cota para cada ciudad a visitar porque la solución al problema requiere que se pase una vez por todas las ciudades, y se vuelva a la ciudad de partida.

Acotación comenzará cuando se encuentre la primera solución, es decir cuando se complete el circuito (secuencia de $n+1$ ciudades).

Resolución de problemas

Algoritmo de Propósito Especial – Agente Viajero

Inicio

1. $LE := ([1]); \quad UB = \infty$

Test de Parada:

2. Si ($LE = ()$) entonces Escribir(“no hay solución”), PARE;

3. $LISTA := Mejor(LE);$

4. $P := Ultimo(LISTA);$

5. Si ($L(LISTA) = n+1$) entonces Escribir(“solución=”, LISTA), PARE;

Ramifica y Acota:

6. Elimina_mejor(LE);

7. Para ($Nodo \in Hijos(P)$)

7.1 $W_L := LISTA;$

7.2 Adiciona_ultimo(Nodo, W_L);

7.3 Si ($f(W_L) < UB$) entonces

7.4 \rightarrow Adiciona_inicio(W_L, LE);

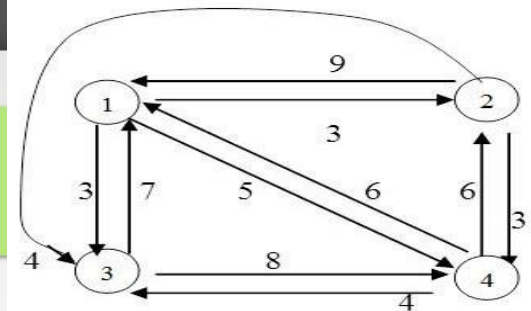
7.5 Si ($L(W_L) = n+1$) entonces $UB = f(W_L);$

Fin-Para

8. Ir Para 2

Resolución de problemas

función de evaluación: distancia recorrida



Its	LE	LISTA	UB
1	((1)-0)	(1)-0	∞
2	((1 2)-3 (1 3)-3 (1 4)-5)	(1 2)-3	∞
3	((1 2 4)-6 (1 2 3)-7 (1 3)-3 (1 4)-5)	(1 3)-3	∞
4	((1 3 4)-11 (1 2 4)-6 (1 2 3)-7 (1 4)-5)	(1 4)-5	∞
5	((1 4 3)-9 (1 4 2)-11 (1 3 4)-11 (1 2 4)-6 (1 2 3)-7)	(1 2 4)-6	∞
6	((1 2 4 3)-10 (1 4 3)-9 (1 4 2)-11 (1 3 4)-11 (1 2 3)-7)	(1 2 3)-7	∞
7	((1 2 3 4)-15 (1 2 4 3) -10 (1 4 3)-9 (1 4 2)-11 (1 3 4)-11	(1 4 3)-9	∞
8	((1 2 3 4)-15 (1 2 4 3) -10 (1 4 2)-11 (1 3 4)-11)	(1 2 4 3) -10	∞
9	((1 2 4 3 1) -17 (1 2 3 4)-15 (1 4 2)-11 (1 3 4)-11)	(1 3 4)-11	17
10	((1 2 4 3 1) -17 (1 2 3 4)-15 (1 4 2)-11)	(1 4 2)-11	17
11	((1 4 2 3)-15 (1 2 4 3 1) -17 (1 2 3 4)-15)	(1 4 2 3)-15	17
12	((1 2 4 3 1) -17 (1 2 3 4)-15)	(1 2 3 4)-15	17
12	((1 2 4 3 1) -17)	(1 2 4 3 1) -17	17

Resolución de problemas

Agente Viajero

La ruta solución es 1-2-4-3-1 con un valor de 17 unidades

El algoritmo de RyA ha proporcionado la ruta de menor suma de distancias para el agente viajero