Práctica 5

VUELTA ATRÁS

RAMIFICACIÓN Y PODA

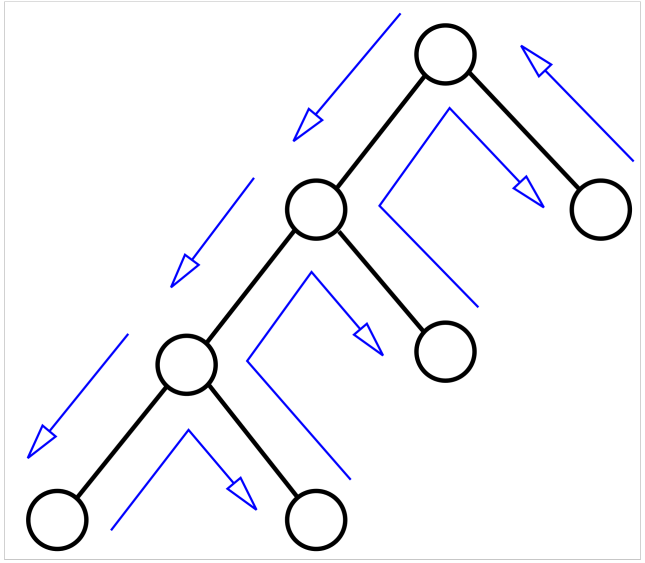
Realizado por Victor Rubia López

CONTENIDO

VUELTA ATRÁS

DESCRIPCIÓN GENERAL

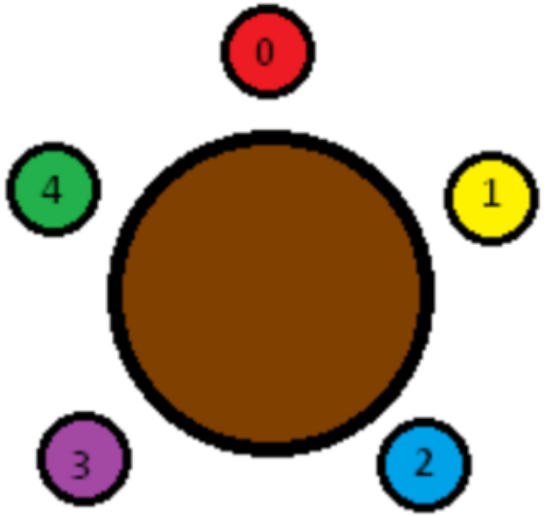
1. **Resolver problemas recursivamente**
2. **Encontrar una solución incrementalmente**
3. **Considerar cada combinación posible**



CENA DE GALA

Análisis Del Problema

1. ***n* Invitados**
2. **Cada invitado tiene un nivel de conveniencia con otros**
3. **Solución: Secuencia de invitados para tener la mayor conveniencia**

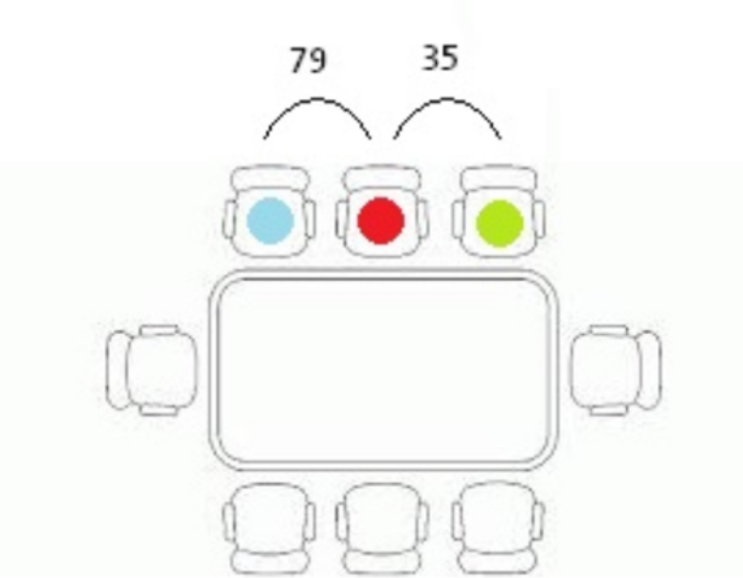


CENA DE GALA

Análisis Del Problema

3 **Representación de la afinidad del invitado rojo con el azul y verde**

4 **El resto de invitados también tienen una afinidad *x* con el rojo**



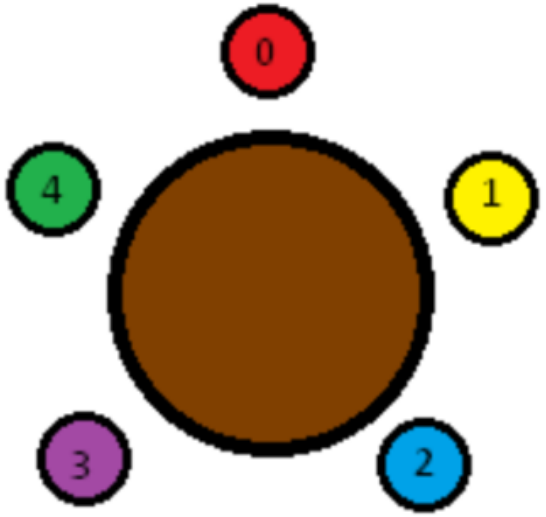
CENA DE GALA

Análisis Del Problema

5 **Ejemplo de invitados a la cena**

Afinidad:

0-1 (30) 1-2 (80) 2-3 (40) 3-4 (1)
0-2 (50) 1-3 (10) 2-4 (20)
0-3 (60) 1-4 (5)
0-4 (70)



CENA DE GALA

Elementos Para Solucionar El Problema

1. **Ejemplo de matriz de conveniencia**

2. **Conveniencia entre el comensal *i* y el *j***

3. **Simétrica y diagonal de 0s**

$$\begin{pmatrix} 0 & 15 & 25 \\ 15 & 0 & 10 \\ 25 & 10 & 0 \end{pmatrix}$$

CENA DE GALA

PSEUDOCÓDIGO

```
función backtracking(sol, sol_parcial, sentados, comensal_actual, nivel)
variables    Matriz,    sol_final[N],    sol_parcial[N],    sentados[N]:=false,
comensal_actual, nivel, val_max:=0

Sentados[comensal_actual]:=true
sol_parcial[nivel-1]:=comensal_actual
para todo i en N hacer
     si sentados[i]==false entonces
          valor_actual=CalcularSolucionActual(sol_parcial)
          backtracking(sol, sol_parcial, sentados, i, nivel+1)
          si nodo_actual==nodo_hoja entonces
               valor_actual:=CalcularSolucionActual()
               si valor_actual es mayor que valor_maximo entonces
                    sol_final:=sol_actual
                    valor_maximo:=valor_actual
               fsi
          sino
               valor_actual = CalcularSolucionActual(sol_parcial)
          fsino
          fsi

          Sentados[i]:=false
     fsi
fpara
```

CENA DE GALA

CASOS DE EJECUCIÓN

$$\begin{pmatrix} 0 & 17 & 97 & 20 & 65 & 36 & 32 \\ 17 & 0 & 21 & 2 & 79 & 53 & 86 \\ 97 & 21 & 0 & 63 & 15 & 20 & 94 \\ 20 & 2 & 63 & 0 & 71 & 49 & 91 \\ 65 & 79 & 15 & 71 & 0 & 54 & 25 \\ 36 & 53 & 20 & 49 & 54 & 0 & 60 \\ 32 & 86 & 94 & 91 & 25 & 60 & 0 \end{pmatrix}$$

```
Para 9 comensales se tarda en ordenar 0.309503
Se sientan así:
0 2 8 3 1 7 6 4 5
(Donde el primero se sienta al lado del último)
Número de nodos explorados:109600
```

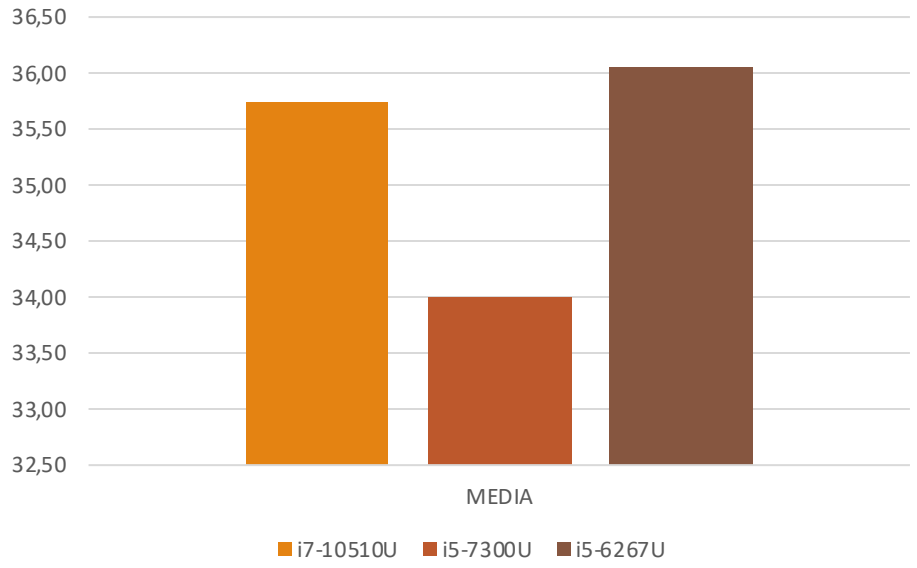
$$\begin{pmatrix} 0 & 95 & 14 & 35 & 22 & 93 & 56 & 3 & 22 & 82 & 89 & 59 \\ 95 & 0 & 57 & 68 & 94 & 51 & 70 & 68 & 57 & 74 & 32 & 41 \\ 14 & 57 & 0 & 51 & 94 & 58 & 70 & 63 & 31 & 93 & 38 & 67 \\ 35 & 68 & 51 & 0 & 2 & 54 & 43 & 31 & 94 & 21 & 100 & 6 \\ 22 & 94 & 94 & 2 & 0 & 10 & 29 & 14 & 22 & 9 & 60 & 83 \\ 93 & 51 & 58 & 54 & 10 & 0 & 63 & 1 & 61 & 40 & 3 & 98 \\ 56 & 70 & 70 & 43 & 29 & 63 & 0 & 78 & 63 & 80 & 34 & 27 \\ 3 & 68 & 63 & 31 & 14 & 1 & 78 & 0 & 61 & 77 & 23 & 40 \\ 22 & 57 & 31 & 94 & 22 & 61 & 63 & 61 & 0 & 94 & 23 & 72 \\ 82 & 74 & 93 & 21 & 9 & 40 & 80 & 77 & 94 & 0 & 51 & 80 \\ 89 & 32 & 38 & 100 & 60 & 3 & 34 & 23 & 23 & 51 & 0 & 48 \\ 59 & 41 & 67 & 6 & 83 & 98 & 27 & 40 & 72 & 80 & 48 & 0 \end{pmatrix}$$

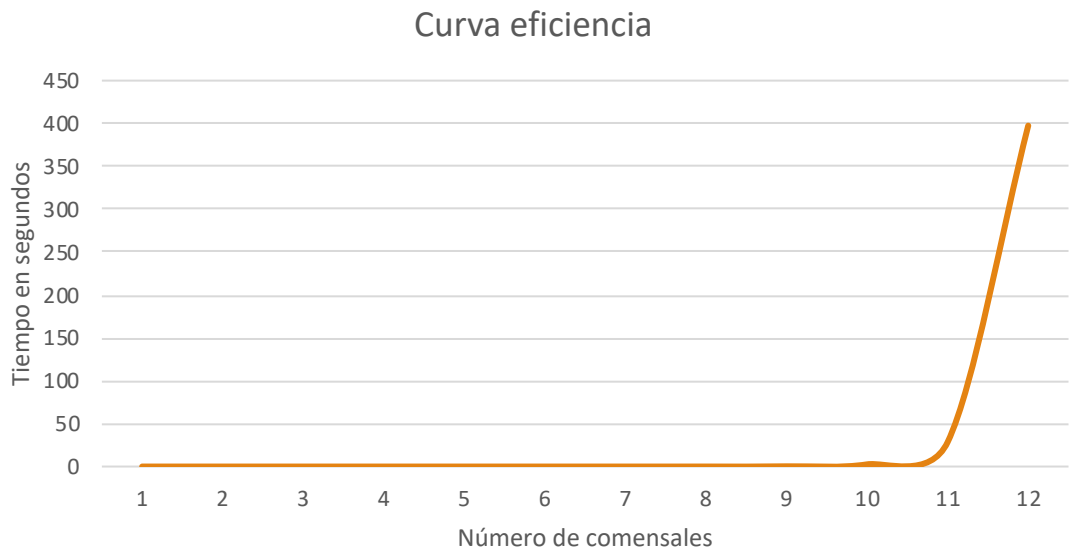
```
Para 12 comensales se tarda en ordenar 443.126
Se sientan así:
0 4 7 10 3 1 6 5 9 8 11 2
(Donde el primero se sienta al lado del último)
Número de nodos explorados:108505111
```

CENA DE GALA

EFICIENCIA EMPÍRICA

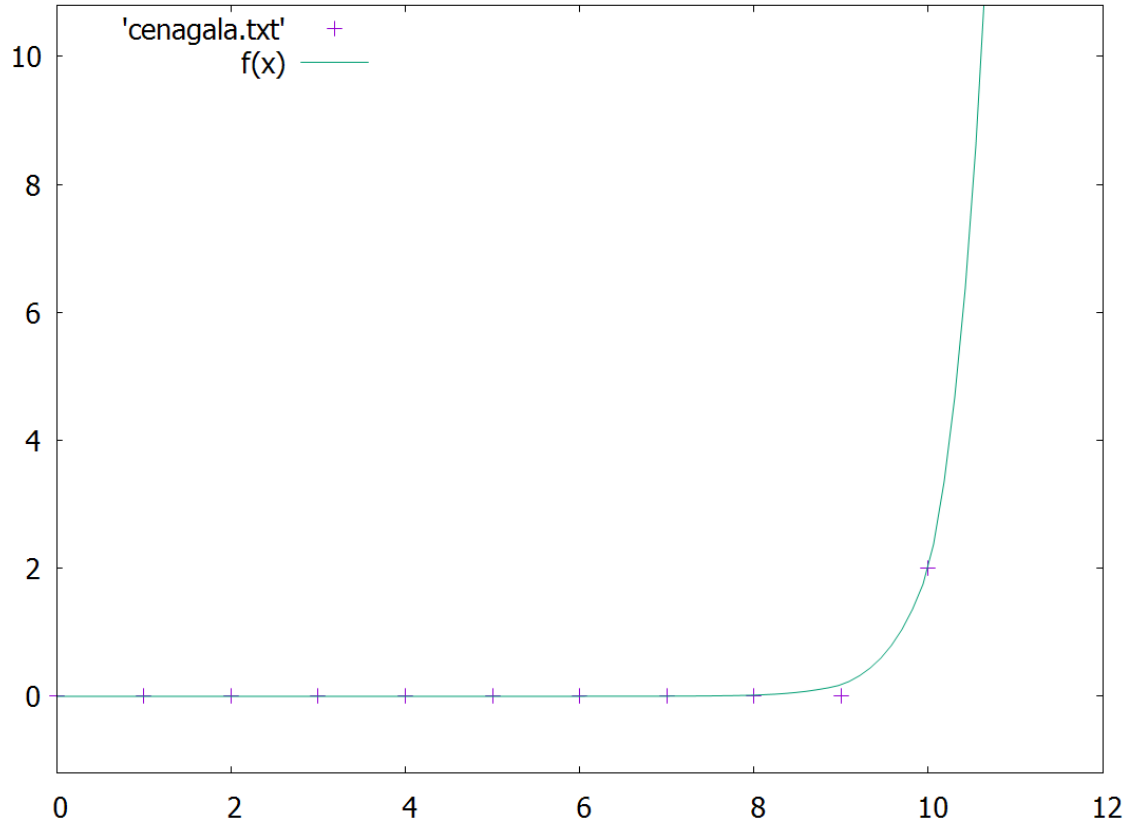
$$O(q(n)n!)$$





CENA DE GALA

EFICIENCIA HÍBRIDA



Ajustamos a

$$f(x) = a \cdot x \cdot x!$$

```
Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                 = 5.46675e-08    +/- 1.564e-09    (2.86%)
```

$$f(x) = 0.0000000546675 \cdot x \cdot x!$$

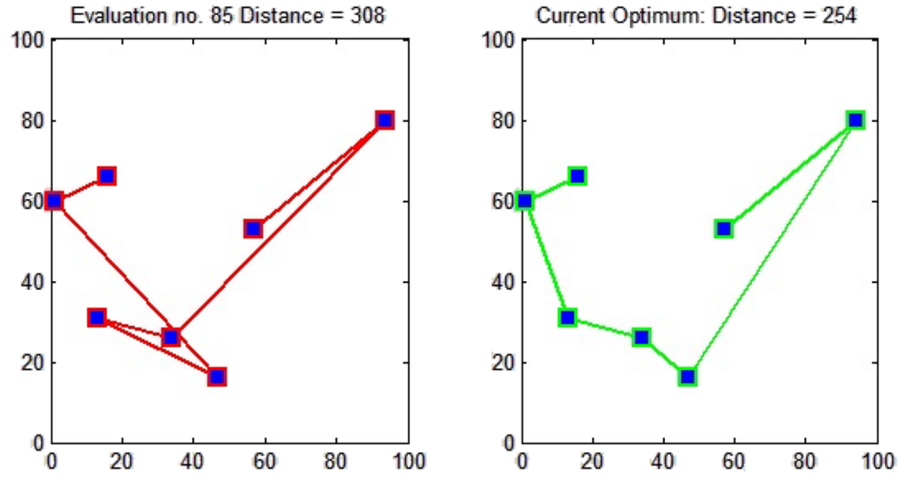
PROBLEMA DEL VIAJANTE DE COMERCIO

ENFOQUE BASADO EN RAMIFICACIÓN Y PODA

1. **Usamos una cota local**

2. **Estimaciones optimistas calculando arcos salientes de la ciudad *i*.**

3. **Criterio LC “más prometedor” para expandir el árbol de búsqueda**



PSEUDOCÓDIGO

PROBLEMA DEL VIAJANTE DE COMERCIO BASADO EN RAMIFICACIÓN Y PODA

1. **Cola con prioridad**
2. **Árboles, nodos vivos y vector de ciudades visitadas**
3. **Si un nodo hoja cuya cota local es menor que la global, actualizamos la cota global y la solución final**

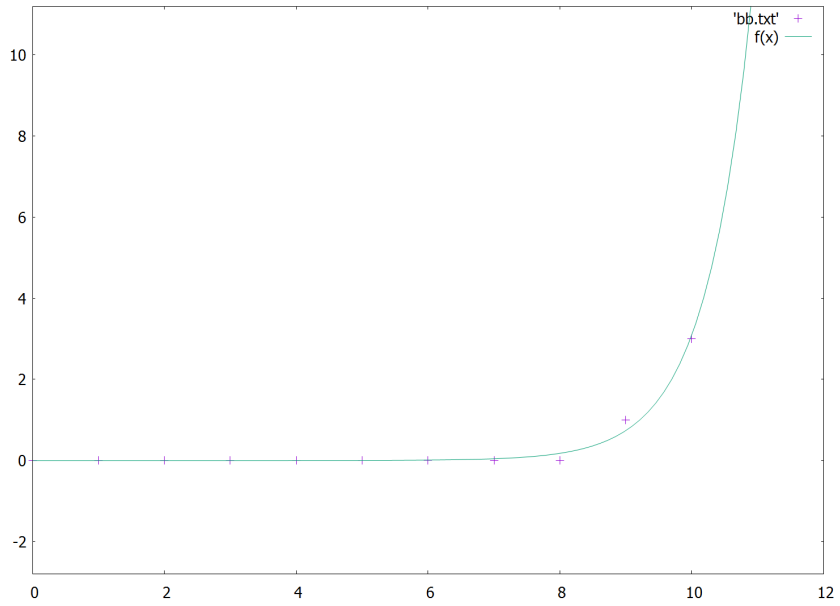
```
función tspbb(matriz_costes, vector_ciudades, vector_dist_min)
variables matriz_costes, vector_ciudades[N], vector_dist_min, cola, sol_final
Nodo n.generarnodosvivos(vector_ciudades[0])
mientras !cola.vacia() hacer
      nodo:=cola.top()
      si EsHoja(nodo) and nodo.cotalocal es mayor que cota_global entonces
            sol_final:= nodo.solucion
            cota_global:=nodo.cotalocal
      fsi
      si nodo.cotalocal es menor que cota_global entonces
            cola.add(nodo.generarhijos())
            cola.push()
      sino
            devuelve solucion_final
      fsino
      fsi
fmientras
```

PROBLEMA DEL VIAJANTE DE COMERCIO

TIEMPOS OBTENIDOS. FUNCIÓN DE EFICIENCIA. RAMIFICACIÓN Y PODA.

att48

$O(2^n)$




```
Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                = 1.23748e-06     +/- 6.53e+05     (5.277e+13%)
b                = 2.07027         +/- 0.218        (10.53%)
c                = 0.550565        +/- 7.612e+11    (1.383e+14%)
```

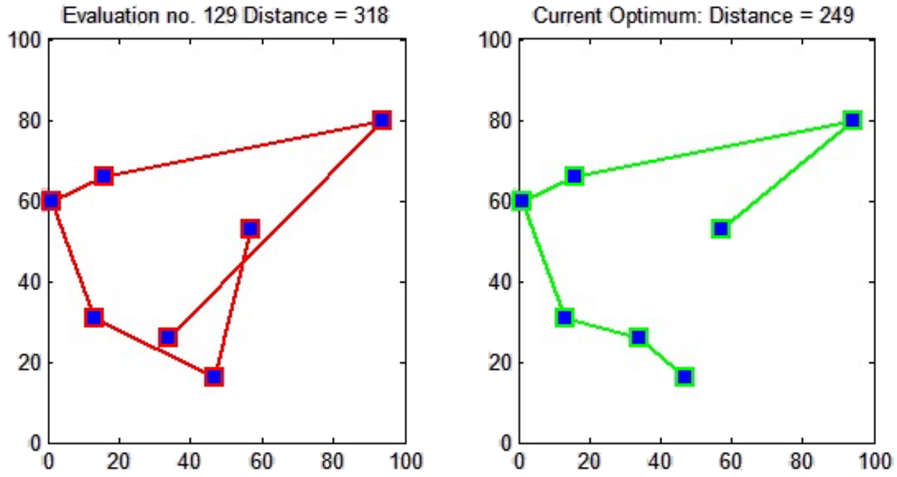
$$f(x) = 0.00000123748 \cdot 2^{2{,}07027x+0{,}550565}$$

*Resultados obtenidos en computador con i7-10510U

PROBLEMA DEL VIAJANTE DE COMERCIO

ENFOQUE BASADO EN VUELTA ATRÁS

1. **Recorremos el árbol en profundidad**
2. **Calculamos cuánto vale en cada nodo la cota local**
3. **Podamos el nodo cuando su cota local sea mayor que la global**



PSEUDOCÓDIGO

PROBLEMA DEL VIAJANTE DE COMERCIO BASADO EN RAMIFICACIÓN Y PODA

1. **Si la ciudad i no ha sido visitada todavía, establecemos su cota local**
2. **Comprobamos que sea menor que la cota global**
3. **Si se cumple, bajaremos un nivel en el árbol**
4. **Si estamos en nodo hoja, cerramos el circuito.**

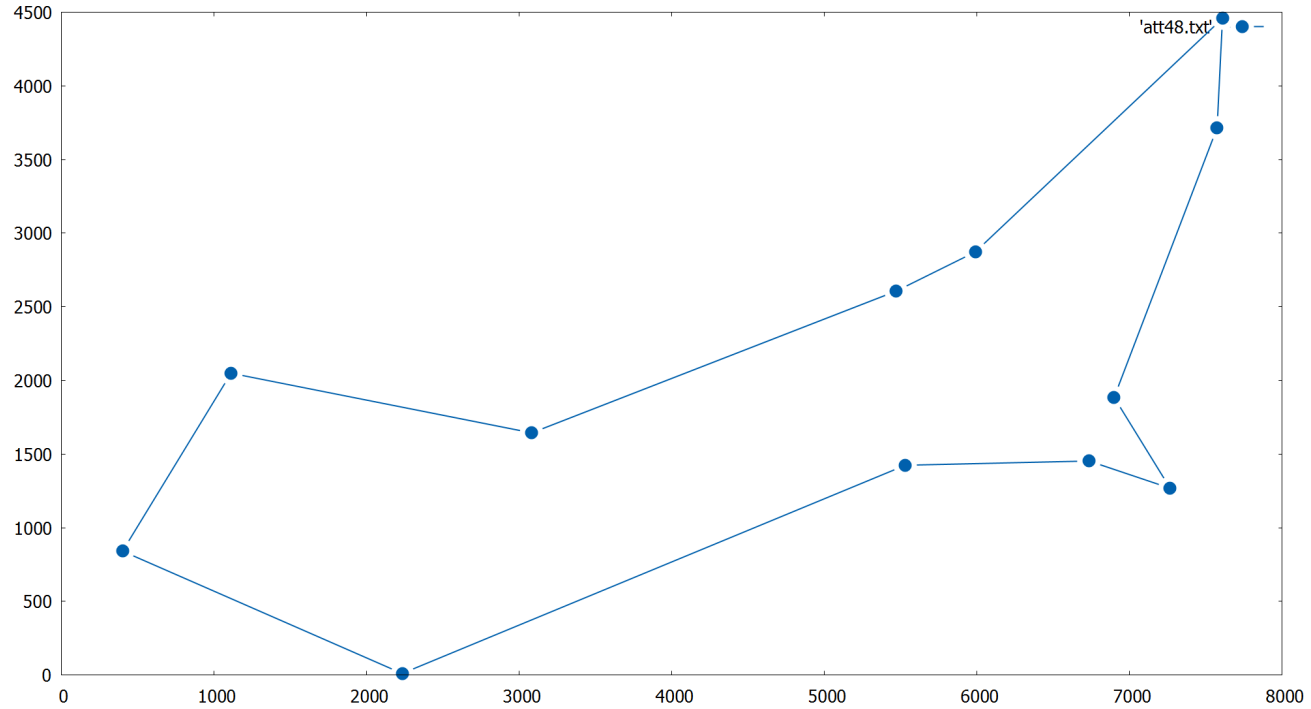
```
función tspbacktracking(sol, sol_parcial, ciudades, ciudad_actual, nivel)
variables matriz, sol_final[N], sol_parcial[N], ciudades[N]:=false,
ciudad_actual, nivel, val_max:=0


ciudades[ciudad_actual]=true
sol_parcial[nivel-1]=ciudad_actual


para i hasta N hacer
      si ciudades[i] es igual que false entonces
            valor_acutal:=calcularSolucionActual(sol_parcial)
            si cotalocal es menor que cotaglobal entonces
                  tspbacktracking(sol, sol_parcial, ciudades, i, nivel+1)
            fsi
            si nodo_actual es igual que nodo_hoja entonces
                  valor_actual:=CalcularSolucionActual(sol_parcial)
                  si valor_actual es mayor que valormaximo entonces
                        sol_final:=sol_actual
                        valor_maximo:=valor_actual
                  fsi
            fsi
            ciudades[i]:=false
      fsi
fpara
```

Algunos Escenarios De Ejecución

att48



Vuelta Atrás

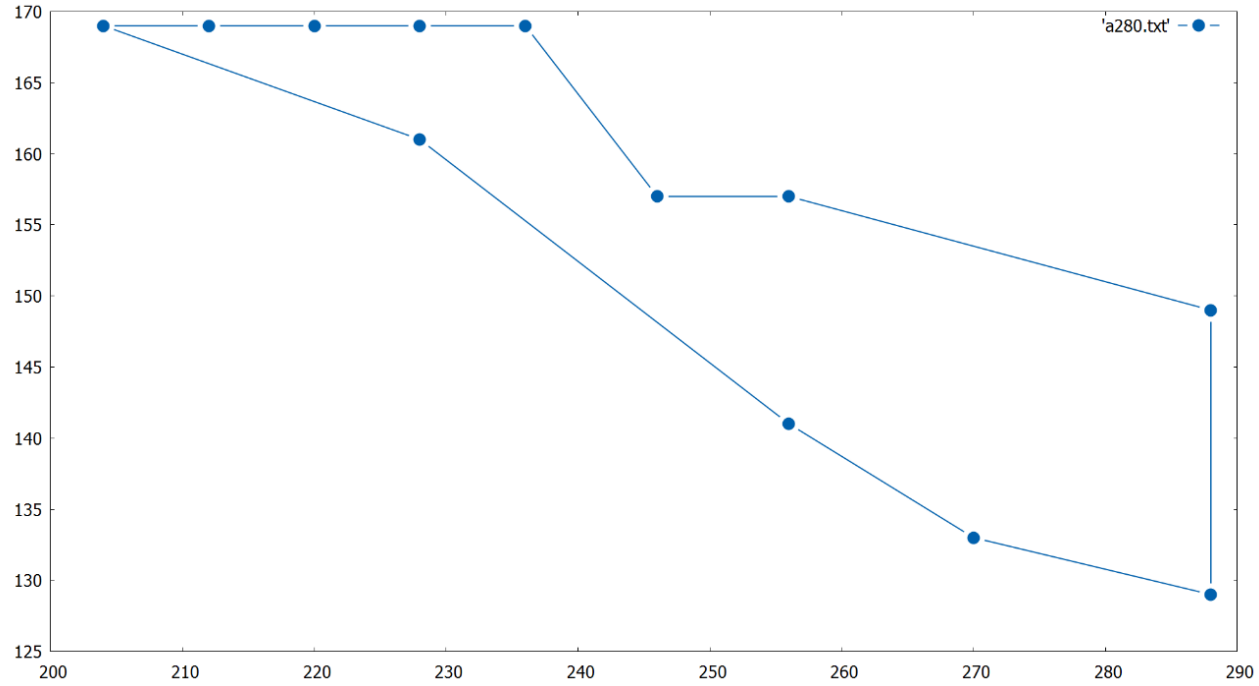
DISTANCIA: 19614.6
TOUR: 1 3 2 4 10 5 11 12 6 7 9 8 1
NODOS TOTALES: 479001600
NODOS PODADOS: 478081610
NODOS EXPLORADOS: 919990
TIEMPO (seg): 0.819056

Ramificación y Poda

DISTANCIA: 19614.6
TOUR: 1 3 2 4 10 5 11 12 6 7 9 8 1
NODOS TOTALES: 479001600
NODOS EXPLORADOS: 120931
NODOS EXPANDIDOS: 442282
NODOS PODADOS: 478880669
TIEMPO (seg): 13.8036

ALGUNOS ESCENARIOS DE EJECUCIÓN

a280



Vuelta Atrás

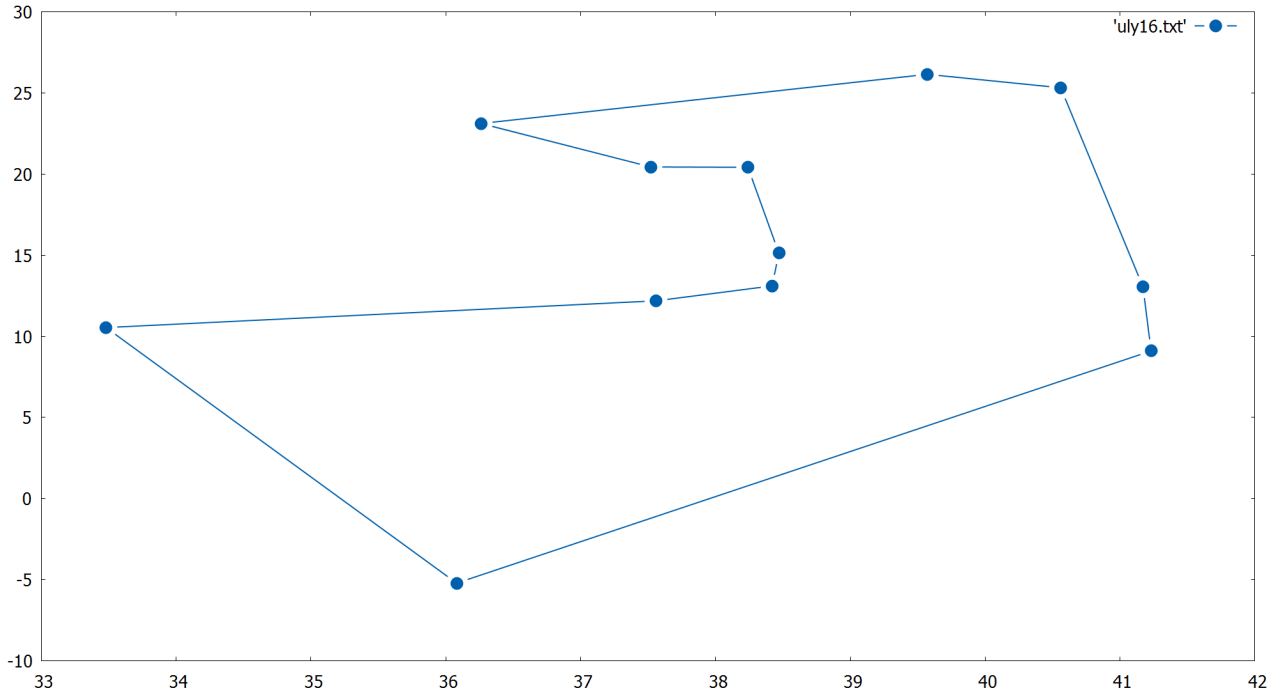
DISTANCIA: 204.876
TOUR: 1 5 6 7 8 10 11 12 9 4 3 2 1
NODOS TOTALES: 479001600
NODOS PODADOS: 478101109
NODOS EXPLORADOS: 900491
TIEMPO (seg): 0.777013

Ramificación y Poda

DISTANCIA: 204.876
TOUR: 1 5 6 7 8 10 11 12 9 4 3 2 1
NODOS TOTALES: 479001600
NODOS EXPLORADOS: 146053
NODOS EXPANDIDOS: 451766
NODOS PODADOS: 478855547
TIEMPO (seg): 15.6944

ALGUNOS ESCENARIOS DE EJECUCIÓN

ulysses16



Vuelta Atrás

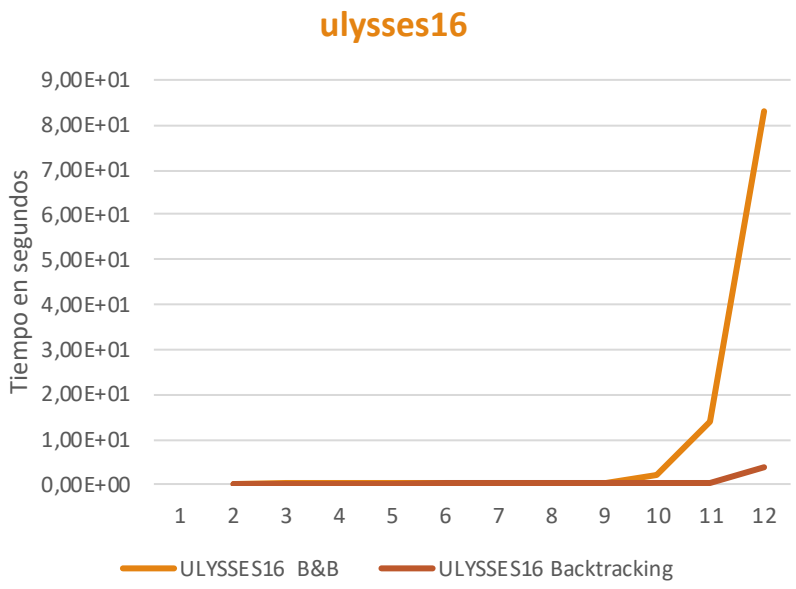
DISTANCIA: 69.8443
TOUR: 1 12 7 6 5 11 9 10 3 2 4 8 1
NODOS TOTALES: 479001600
NODOS PODADOS: 474474230
NODOS EXPLORADOS: 4527370
TIEMPO (seg): 3.74082

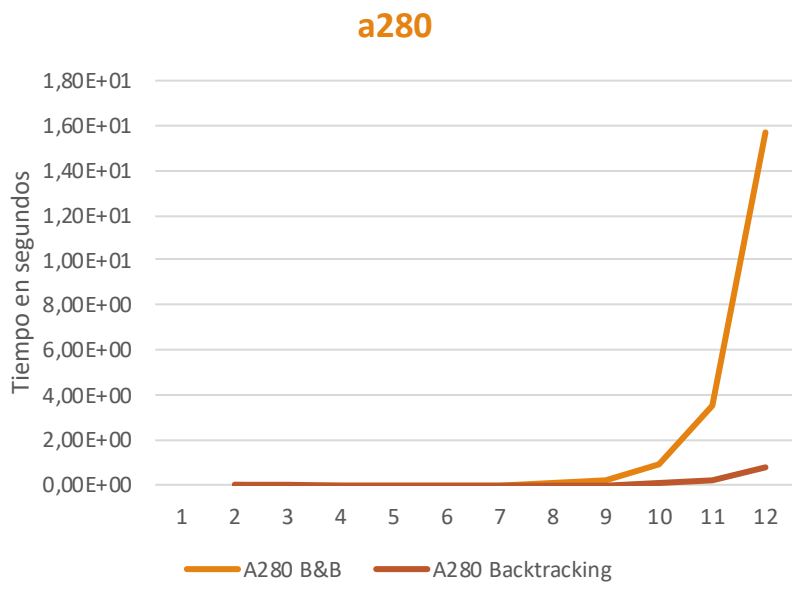
Ramificación y Poda

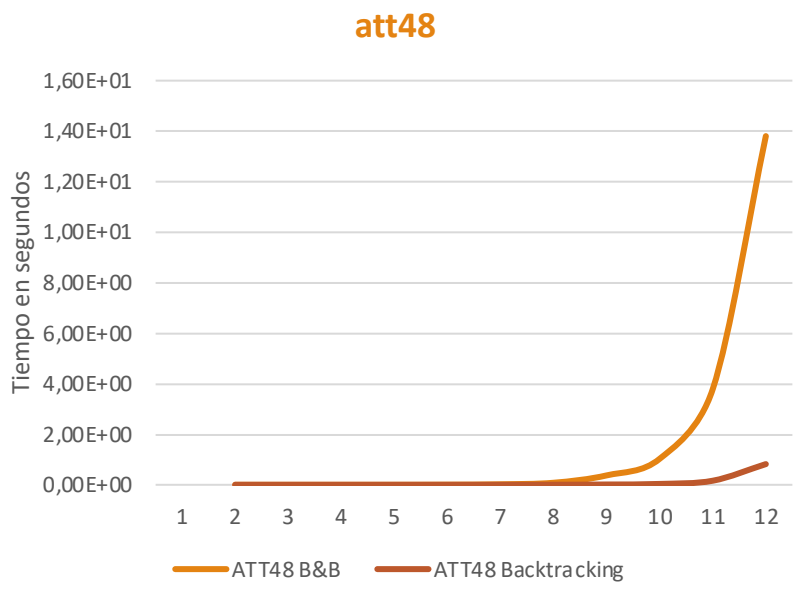
DISTANCIA: 69.8443
TOUR: 1 12 7 6 5 11 9 10 3 2 4 8 1
NODOS TOTALES: 479001600
NODOS EXPLORADOS: 716897
NODOS EXPANDIDOS: 2233532
NODOS PODADOS: 478284703
TIEMPO (seg): 83.0159

PROBLEMA DEL VIAJANTE DE COMERCIO

TIEMPOS OBTENIDOS. COMPARATIVA GRÁFICA

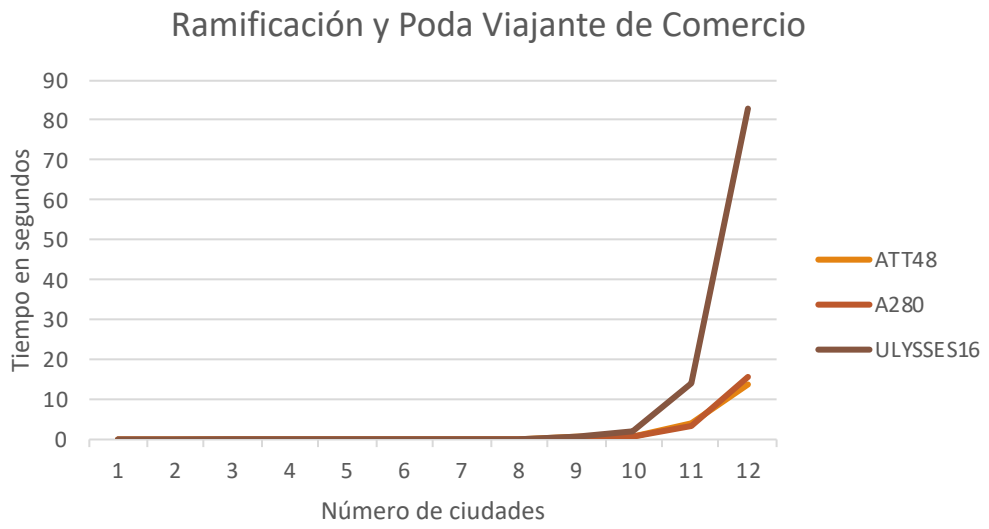


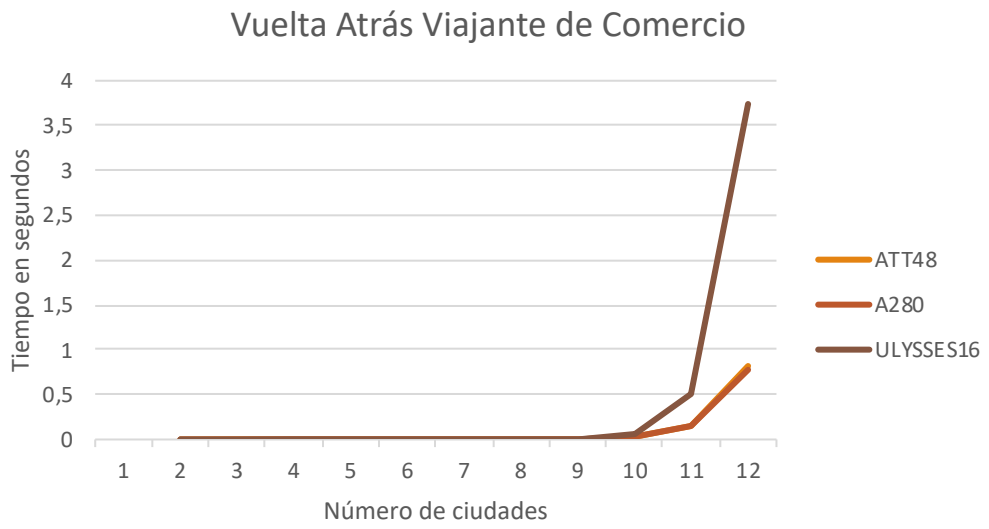




PROBLEMA DEL VIAJANTE DE COMERCIO

TIEMPOS OBTENIDOS. COMPARATIVA GRÁFICA





PROBLEMA DEL VIAJANTE DE COMERCIO

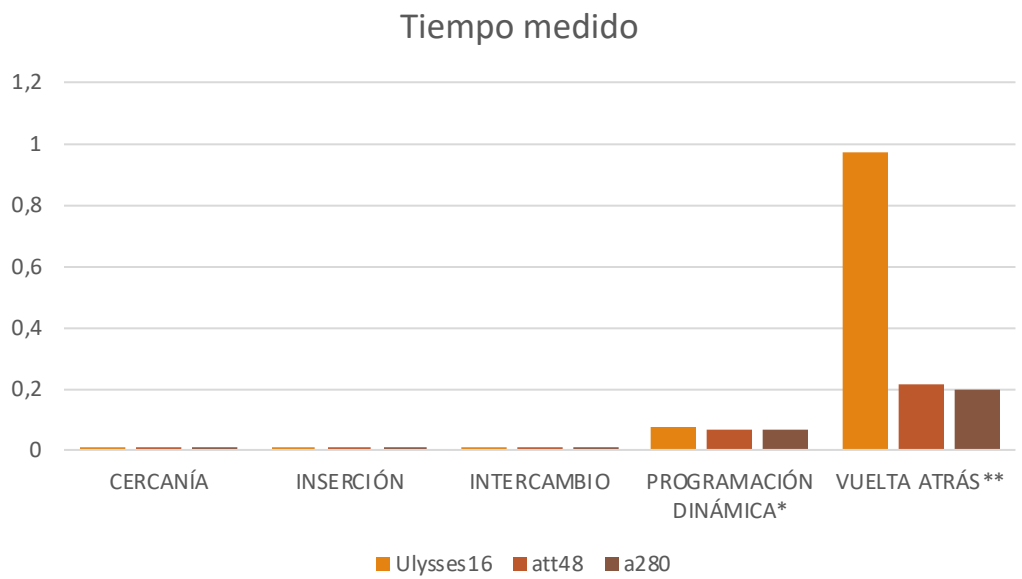
COMPARACIÓN TIEMPOS OBTENIDOS

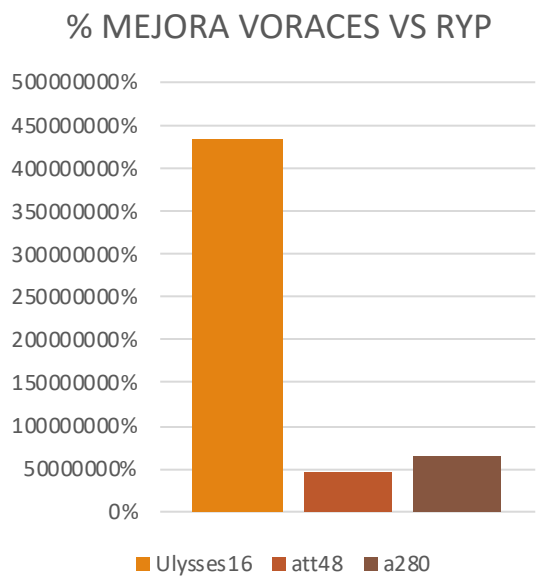
TIEMPO										
	CERCANÍA	INSERCIÓN	INTERCAMBIO	PROGRAMACIÓN DINÁMICA*	RAMIFICACIÓN Y PODA**	VUELTA ATRÁS**	% MEJORA VORACES VS RYP	% MEJORA VORACES VS V.A.	% MEJORA PROG. DINÁMICA VS. RYP	% MEJORA PROG. DINÁMICA VS V.A.
Ulysses16	0,000024	0,000015	0,000081	0,070782	65,2494	0,96732	434996000%	6448800%	92183,6%	1366,6%
att48	0,000024	0,000036	0,000095	0,067714	10,8885	0,213248	45368750,0%	888533,3%	16080,1%	314,9%
a280	0,000032	0,000019	0,000081	0,066633	12,4194	0,199173	65365263,2%	1048278,9%	18638,5%	298,9%
hasta 12 nodos										

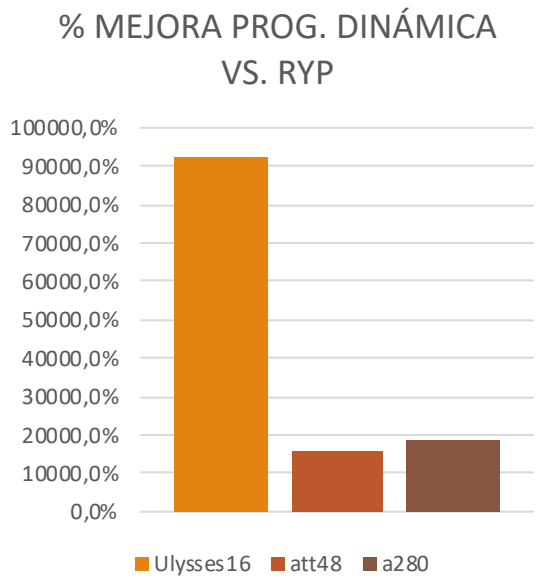
*Resultados obtenidos en computador con i7-8700B

PROBLEMA DEL VIAJANTE DE COMERCIO

COMPARACIÓN TIEMPOS OBTENIDOS. GRÁFICAS







*Resultados obtenidos en computador con i7-8700B

PROBLEMA DEL VIAJANTE DE COMERCIO

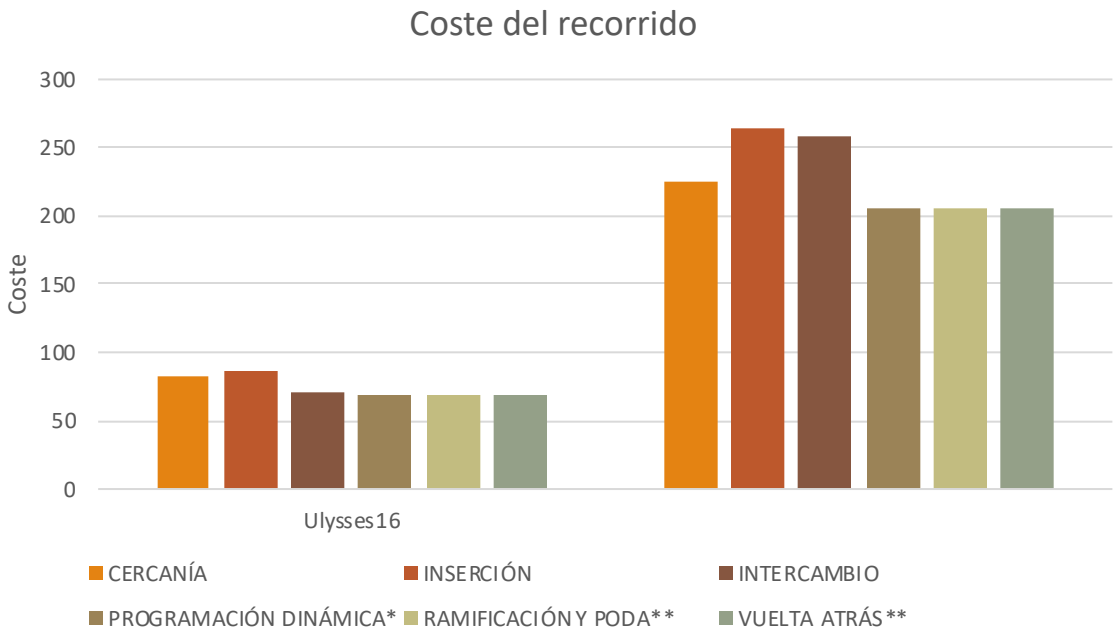
COMPARACIÓN COSTES OBTENIDOS

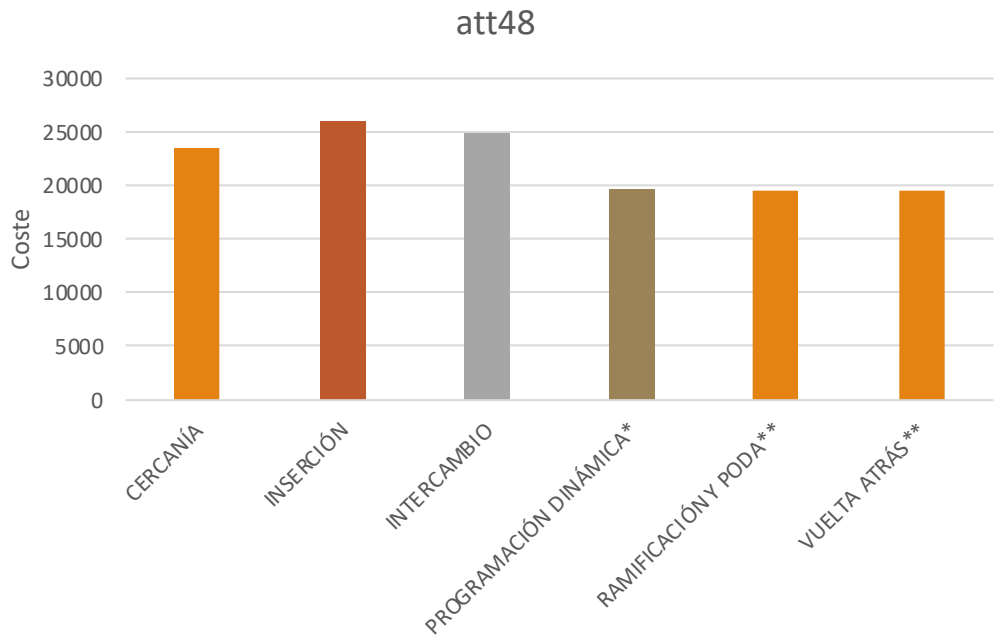
	COSTE									
	CERCANÍA	INSERCIÓN	INTERCAMBIO	PROGRAMACIÓN DINÁMICA*	RAMIFICACIÓN Y PODA**	VUELTA ATRÁS**	% MEJORA VORACES VS RYP	% MEJORA VORACES VS V.A.	% MEJORA PROG. DINÁMICA VS. RYP	% MEJORA PROG. DINÁMICA VS V.A.
Ulysses16	83	86	71	69,84	69,84	69,84	9837%	9837%	100,00%	100,00%
att48	23466	25977	24860	19614,60	19614,60	19614,60	8358,7%	8358,7%	100,00%	100,00%
a280	225	265	258	205	205	205	9105,6%	9105,6%	100,00%	100,00%
*= hasta 22 nodos										

*Resultados obtenidos en computador con i7-8700B

PROBLEMA DEL VIAJANTE DE COMERCIO

COMPARACIÓN COSTES OBTENIDOS. GRÁFICAS

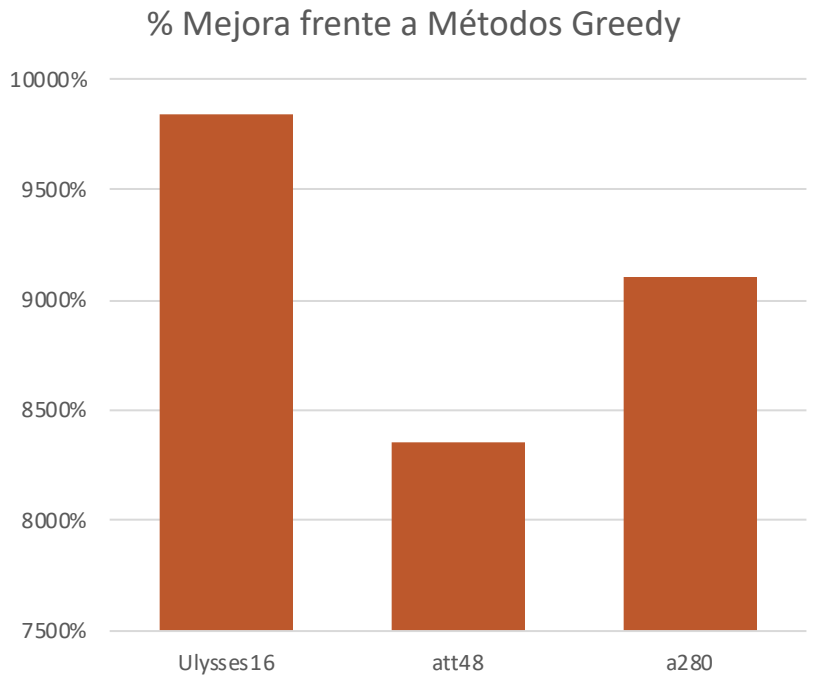




*Resultados obtenidos en computador con i7-8700B

PROBLEMA DEL VIAJANTE DE COMERCIO

COMPARACIÓN COSTES OBTENIDOS. GRÁFICAS



*Resultados obtenidos en computador con i7-8700B

CONCLUSIÓN

- Para qué tipo de problemas usar Vuelta Atrás y Ramificación y Poda

- Obtener solución óptima

- Incremento tiempo computacional frente a otros métodos

Gracias