

Práctica 4-Primera Parte: Cena de gala

Francisco Carrillo Pérez, Borja Cañavate Bordons,
Miguel Porcel Jiménez, Jose Manuel Rejón Santiago, Jose Arcos Aneas

18 de mayo de 2016

Índice

1. Introducción	3
2. Elementos de la solución al problema	3
2.1. Representación de la compatibilidad	3
2.2. Representación de la solución	3
2.3. Solucion parcial	3
2.4. Restricciones explícitas	3
2.5. Restricciones implícitas	3
3. Pseudocódigo	3
4. Eficiencia	4
4.1. Eficiencia Híbrida	5

Índice de figuras

4.1. algoritmo backtracking	4
4.2. comparación x^{**8} vs algoritmo backtracking	5

Índice de tablas

4.1. Ef híbrida obtenida	5
------------------------------------	---

1. Introducción

El objetivo de esta práctica es diseñar un algoritmo Backtracking, que resuelva uno de los cinco problemas de la práctica y realizar un estudio empírico de su eficiencia.

Se desea sentar a N invitados alrededor de una mesa, de manera que cada invitado tendrá a su lado a otros dos. Cada par de invitados tiene un nivel de compatibilidad. Se desea maximizar la compatibilidad de estos comensales.

2. Elementos de la solución al problema

Dada una matriz $M[i][j]$ Mantenemos en la matriz la afinidad entre el comensal i y el comensal j

$$\begin{pmatrix} 0 & 30 & 15 \\ 30 & 0 & 20 \\ 15 & 20 & 0 \end{pmatrix}$$

2.1. Representación de la compatibilidad

La entrada será una matriz simétrica de valores aleatorios con la diagonal de 0s.

2.2. Representación de la solución

Vector de longitud igual al número de invitados (N), en que cada posición guarda el valor del invitado que se sienta en la posición i .

2.3. Solución parcial

Solución parcial al problema de tamaño menor que N .

2.4. Restricciones explícitas

Los valores que puede tomar la solución son los enteros de 1 a N . Donde N es el número total de invitados.

2.5. Restricciones implícitas

Estas restricciones son las que determinan si una función parcial puede llevarnos a una solución del problema. Si supera un umbral.

3. Pseudocódigo

Require: Matriz, $S_final[N]$ $S_parcial[N]$ $Sentados[N]=false$ comensal_actual, nivel, valor_maximo=0;
Funcion($S, S_parcial, Sentados, comensal_actual, nivel$):
 $Sentados[comensal_actual]=true$;

```

S_parcial[nivel - 1]=comensal_actual;
for i to N do
  if Sentados[i]==false then
    valor_actual = CalcularSolucionActual(S_parcial);
    Funcion(S,S_parcial,Sentados,i,nivel+1);
  if nodo_actual == nodo_hoja then
    valor_actual = CalcularSolucionActual()
    if valor_actual mayor que valor_maximo) then
      S_final = S_Actual
      valor_maximo = valor_actual
    end if
  end if
  Sentados[i] = false;
end if
end for

```

4. Eficiencia

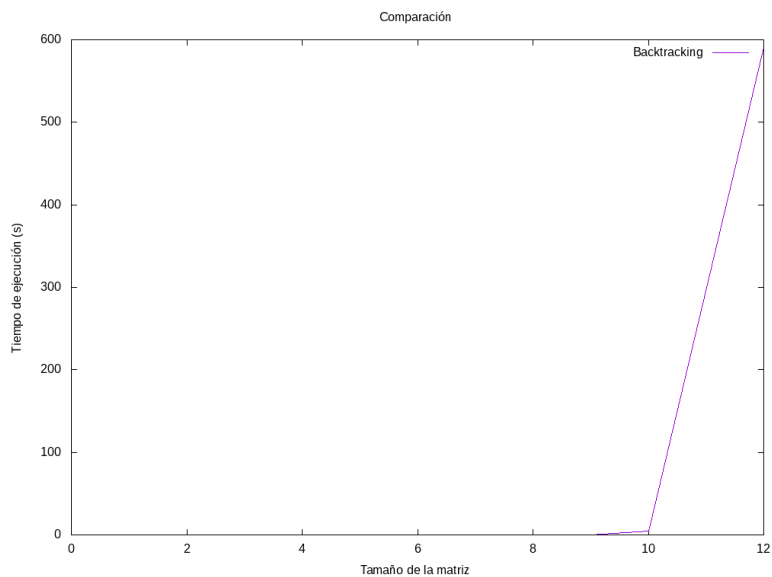


Figura 4.1: algoritmo backtracking

4.1. Eficiencia Híbrida

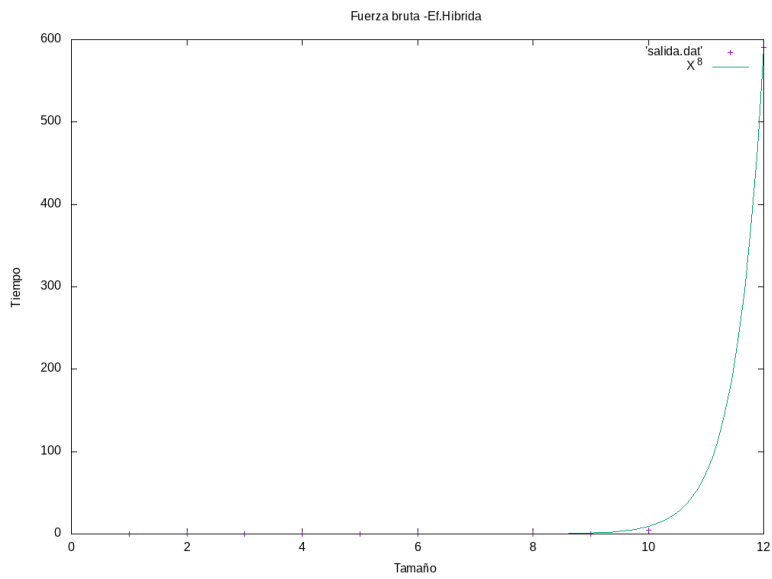


Figura 4.2: comparación x^{**8} vs algoritmo backtracking

Ajuste con X^{**8}

Tabla 4.1: Ef híbrida obtenida

Final set of parameters	Asymptotic Standard Error
$a0 = 8.59555e-09$	$+/- 2.326e-11$ (0.2706 %)