



Algorítmica práctica 5: Backtracking y Branch & Bound

Germán Castilla López

Jorge Gangoso Klöck

Pedro Morales Leyva

Clara M^a Romero Lara

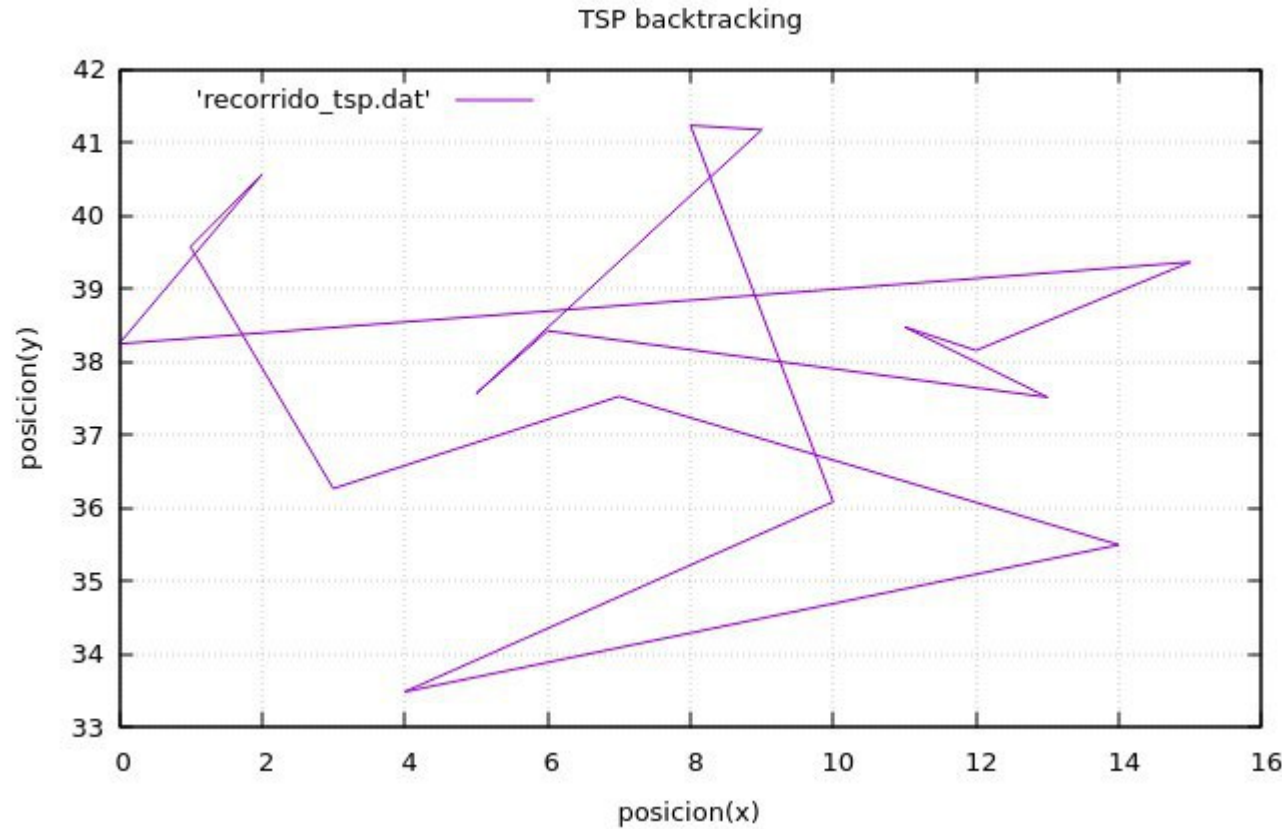
TSP: Backtracking

Pseudocódigo

```

buscarRuta(posActual){
    if(fuera de la matriz)
        res = false
    else
        posActual = parte de la ruta
        rutaFinal.add(posActual)
        if(es el final)
            res = true
        else
            res = true
            if(!buscaRuta(vecinos))
                res = false
    return res
}

```



TSP: Branch & Bound

Pseudocódigo

```
TSP_BB()
{
  Prio_queue Vivos <- inicial //Pila en el caso de backtracking
  Cota_Global <- TSP_Cercanía
  Cota_Local <- NULL
  Ruta_solucion <- NULL

  mientras (vivos no vacío) hacer
  {
    actual <- Vivos.primerO
    Vivos.primerO

    Cota_Local <- mejor camino posible desde actual, idílico

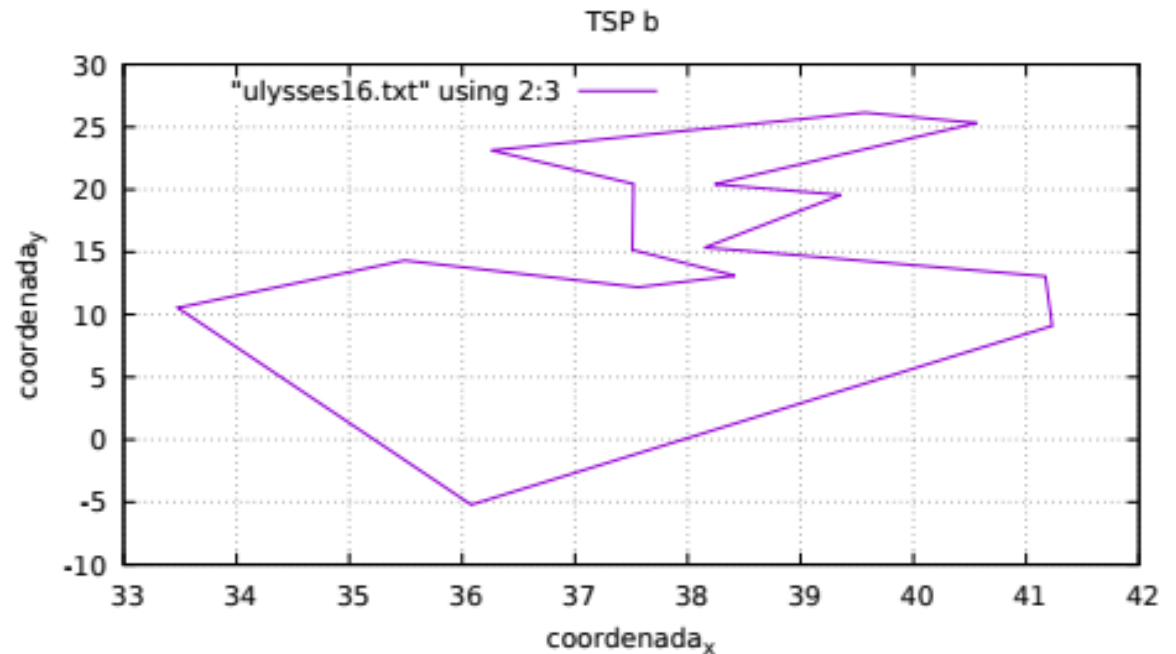
    si fin de ciclo
    {
      si ruta_actual < Cota_Global
      {
        Cota_Global <- ruta_actual
        Ruta_solucion <- ruta_actual
      }
    }

    si criterio de poda (Cota_Local > Cota_Global)
    {
      podar() //No explorar el nodo actual y por tanto ninguna de sus ramificaciones
    }

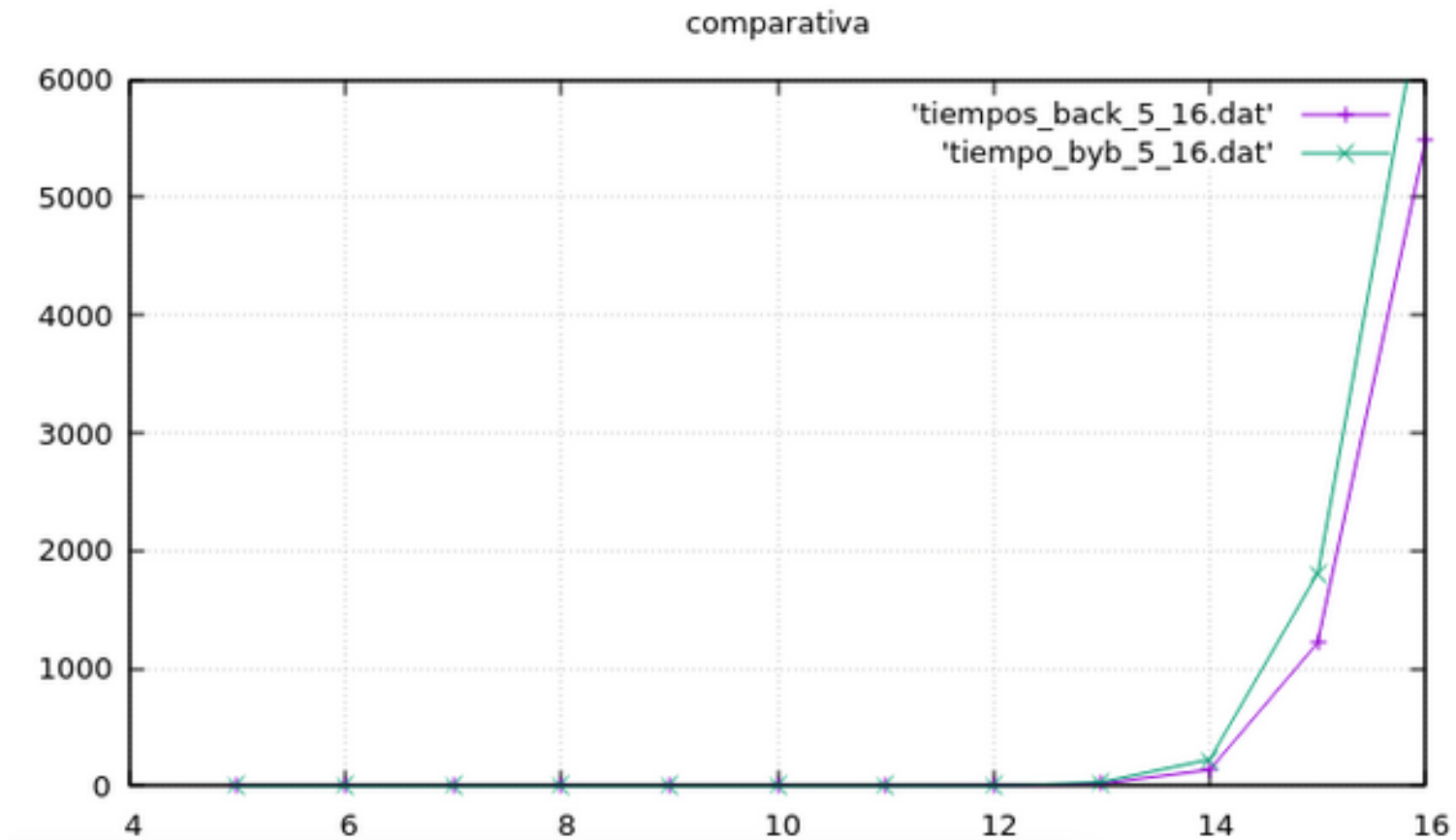
    si no criterio de poda ni fin de ciclo
    {
      Vivos <- actual.generarHijos()
    }
  }

  devolver Ruta_solucion
}
```

3



TSP: Comparación de tiempos de Backtracking y Branch & Bound



Laberinto: Supuesto de ejecución

Pseudocódigo

```
buscarRuta(posActual){  
  if(fuera de la matriz)  
    res = false  
  else  
    posActual = parte de la ruta  
    rutaFinal.add(posActual)  
    if(es el final)  
      res = true  
    else  
      res = true  
      if(!buscaRuta(vecinos))  
        res = false  
  return res
```

Ruta completa

```
0101010  
0100000  
0101110  
0100010  
0101111  
0000000  
0101010
```

Laberinto generado

```
0, 0  4, 2  2, 6  
1, 0  3, 2  3, 6  
2, 0  2, 2  0, 6  
3, 0  1, 2  3, 3  
4, 0  0, 2  3, 4  
5, 0  1, 3  5, 3  
6, 0  1, 4  5, 4  
5, 1  0, 4  5, 5  
5, 2  1, 5  5, 6  
6, 2  1, 6  6, 6
```

Laberinto resuelto

```
2131313  
2133333  
2131113  
2133313  
2131111  
2222222  
3131312
```