

Práctica 4

Backtracking y Branch and Bound

Alumno: Jorge Valenzuela García

1. Voraz

Explicación:

Para hallar la solución a este ejercicio he realizado un bucle, el cual compara en cada iteración una persona con el resto, emparejándola con aquel que tenga menos discrepancia entre el resto de personas.

Mi método tiene una particularidad, y es que, cuando crea una pareja, saca del bucle a la pareja también, es decir, que si en la primera iteración, (persona 1) se le asigna la persona 4, ésta ya no se comprobará en las sucesivas iteraciones, por lo que, suponiendo que sean 6 personas, nos quedarían por analizar a las personas n.º 2, 3, 5 y 6.

Para que fuese como en el ejercicio lo único que habría que hacer es borrar del código la sentencia que agrega a un vector de personas comprobadas a la pareja seleccionada, de este modo, para cada persona se le asociaría su mejor pareja.

El resultado obtenido es el siguiente:

*La pareja de 0 es 4
La pareja de 1 es 3
La pareja de 2 es 5
Discrepancia: 48
Tiempo: 1.1157e-05*

2. Backtracking simple y con poda:

Explicación:

Para comenzar hay que explicar que para encontrar la solución a este ejercicio es necesario usar un árbol permutacional, por lo que me he decantado por buscar en profundidad.

Bien para empezar, lo que el código hace es esperar a que llegue la primera permutación que tenga el mismo tamaño que el número de personas que hay en el archivo grupo.txt que hayamos seleccionado. Por ejemplo si estamos tratando con 6 personas, la primera permutación que nos llegue podría ser 5,6,2,1,3,4, de modo que establecemos la discrepancia total que hay entre esas

parejas (las parejas serían 5-6 , 2-1 y 3-4), de modo que cuando tenemos este caso guardamos la discrepancia.

Bien ahora viene la diferencia entre el backtracking simple y el backtracking con poda.

En el simple, lo único que espera al algoritmo es otra permutación de tamaño igual a las personas del archivo, como al comienzo, y compara la discrepancia que se obtiene de dicha permutación con la anterior. Si es mejor la discrepancia actual, la actualiza y guarda como mejores parejas las que indica la permutación actual.

Al acabar de explorar todos los nodos tendremos a las parejas con menor discrepancia entre ellas.

Con respecto a la técnica backtracking con poda, la diferencia es que una vez obtenemos las primeras parejas y con ello la primera discrepancia, empezamos a comprobar para cada nueva permutación creada a partir de ese momento, si las parejas que están creando tienen una discrepancia mayor que las encontradas anteriormente, de modo que si encontramos un par de parejas (en el caso de tener 6 personas) que ya acumulan mas discrepancia entre ellas que las encontradas previamente, pasamos a usar backtracking. De este modo ahorramos recursos al no seguir bajando y buscando por nodos sabemos de antemano que van a ser peores.

En el caso de tener una permutación de tamaño impar, lo que se hace es comprobar para esa persona libre, su discrepancia mínima con todas las otras personas libres que aún no han sido emparejadas, de modo que antes de crear la pareja en sí ya sabríamos si la discrepancia que vamos a tener va a ser mejor o peor que la actual.

Los resultados son los siguientes:

(tener en cuenta que cuando encuentro la pareja de un persona, marco a esa pareja como buscada, por lo que ya no se suma la discrepancia de esa persona de nuevo. Esto lo que hace es que la discrepancia sea la mitad).

Resultados de backtracking simple:

El numero total de secuencias sin restricciones extras: 1956

Nodos explorados: 78

Pareja de: 3 es: 6

Pareja de: 1 es: 5

Pareja de: 4 es: 2

Discrepancia: 35

Tiempo medio por nodo: 1.65191e-06

Resultados de backtracking con poda:

El numero total de secuencias sin restricciones extras: 1956

Nodos explorados: 60

Pareja de: 3 es: 6

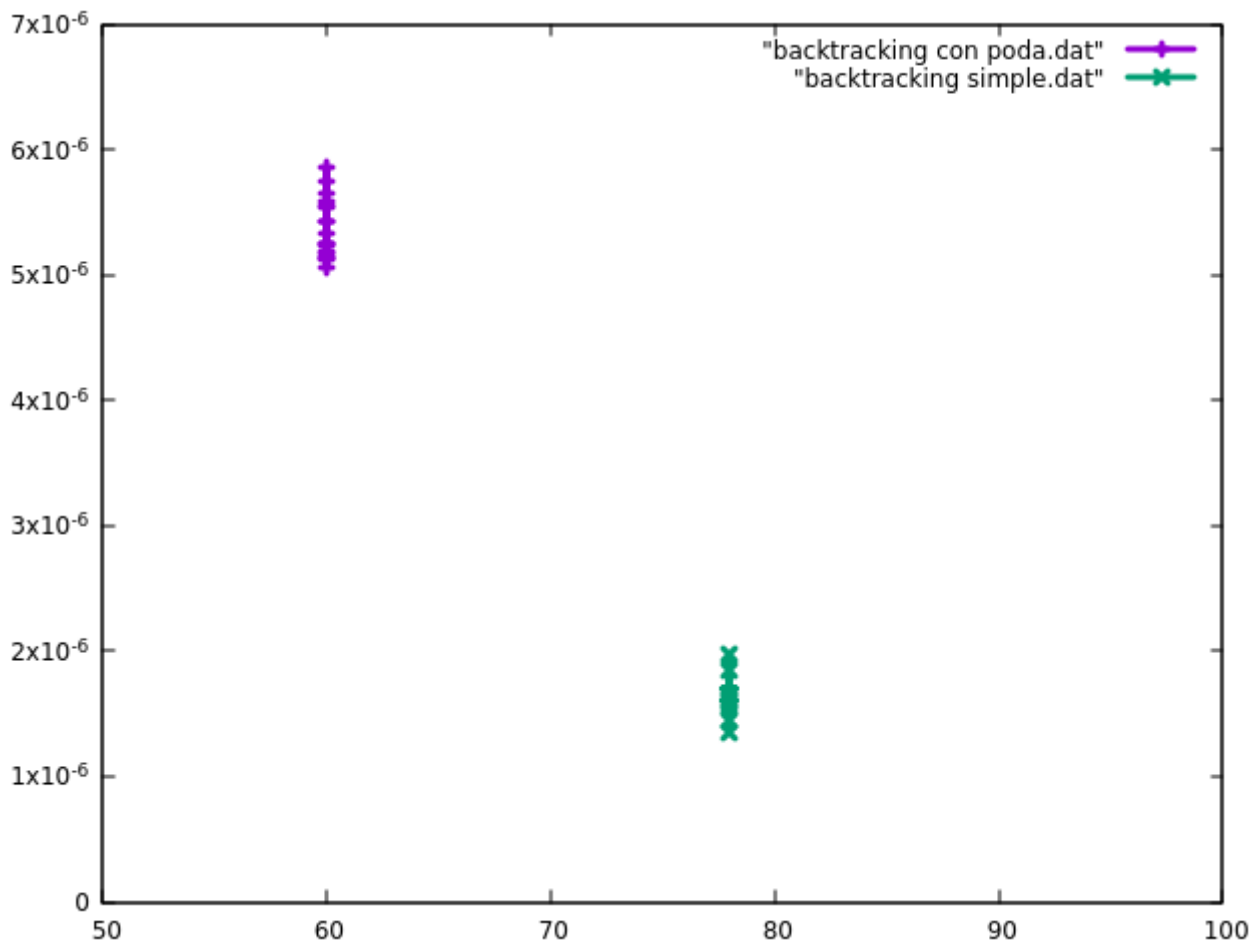
Pareja de: 1 es: 5

Pareja de: 4 es: 2

Discrepancia: 35

Tiempo medio por nodo: 5.13948e-06

Gráfica:



Podemos ver que para la técnica del **backtracking simple** pasa por todos los nodos (78), y el tiempo medio de trabajo en un nodo va desde 1×10^{-6} a 2×10^{-6} segundos, mientras que con la técnica **backtracking con poda** vemos que trabaja en 60 nodos para encontrar solución, además de que el tiempo medio de trabajo en cada nodo ronda desde los 5×10^{-6} a los 6×10^{-6} segundos.

En términos generales, si tendemos los nodos al infinito podemos suponer que tiene mejor resultado el backtracking con poda, ya que aunque tarde algo más, el hecho de poder podar nodos es algo que en árboles muy grandes lleva un gran ahorro de tiempo y recursos.

3. Branch and Bound

Explicación:

Para conseguir dar solución a este ejercicio deberíamos de usar un árbol permutacional, con la diferencia que ahora lo que debemos hacer es ir primero por el camino que tenga una discrepancia estimada menor, es decir, el mejor nodo del nivel en el que nos encontremos.

De este modo estamos encaminando al algoritmo por el lugar que tiene más probabilidad de ser el que lleve a la solución óptima además de establecer unas podas que se producen al buscar por un camino que tiene una discrepancia estimada mayor que la discrepancia actual.

Resultados:

He tenido problemas a la hora de trabajar con esta técnica usando el árbol permutacional, ya que no he podido extraer una buena solución, debido a que la solución que quería aportar es la siguiente. Comenzamos en el nivel 1 del algoritmo, nos quedamos con el mejor nodo, y generamos los hijos de éste, así consecutivamente hasta llegar a la primera permutación de tamaño igual a las personas.

Acto seguido guardaríamos la discrepancia y, pasaríamos a comprobar si alguna de las ramas del árbol puede llegar a tener una discrepancia estimada menor que la conseguida.

La comprobación de las ramas se haría en el orden que dicta la mejor discrepancia estimada.

Con respecto a la gráfica final de comparación entre las técnicas backtracking simple, backtracking con poda, y branch and bound no puedo realizarla debido a que no he podido obtener resultados para la técnica branch and bound, aun así en el apartado 2 aporté los valores conseguidos de la técnica backtracking simple y con poda, donde se puede ver los nodos por los que pasa y el tiempo que tarda.