Práctica Algoritmia Básica

El problema del viajante de comercio

Samuel Torres Fau (NIP: 780505)
Juan Rodríguez Gracia (NIP: 805001)

Algoritmia Básica
Grado en Ingeniería Informática

Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza
Curso 2020/2021

Índice

1. Introducción

El objetivo de esta práctica es aplicar varios distintos tipos algoritmos al problema del viajante de comercio (Travelling Salesman Problem).
Los esquemas empleados son: Algoritmo de fuerza bruta, Algoritmo voraz, Algoritmo de programación dinámica y Algoritmo de ramificación y poda.

2. Coste estimado en tiempo y memoria

2.1. Fuerza bruta

Para implementar este algoritmo hemos realizado las diferentes permutaciones del camino comprobando cual es el de menor coste.

2.2. Algoritmo Voraz

Para implementar este algoritmo se ha seguido la metodología de búsuqueda local: Dado un nodo, escoge el siguiente si no está presente en el camino recorrido y la arista que los une es la de menor coste.

2.3. Programación Dinámica

Para la implementacion de la matriz dinamica hemos utilizado un mapa que contiene como clave un conjunto y como valor asociado el indice de la matriz de adyaciencia.

2.4. Ramificación y poda

Los costes teóricos en tiempo y espacio son parecidos a los de programación dinámica. Sin embargo, esto ocurre para casos que desfavorecen por completo al algoritmo. En la realidad los costes obtenidos son sensiblemente mejor como puede verse en la figura 2. Un ejemplo es de esto puede verse reflejado en la misma tabla, para un problema de dim = 26, en un caso desborda memoria mientras que en el otro lo resuelve relativamente rápido.
El algoritmo se ha desarrollado empleando programación orientada a objetos, y almacenando para cada nodo su matriz reducida correspondiente. Dado que el límite en cuanto al tamaño de matriz viene dado por desbordamiento de memoria, podrían hacerse cambios para que esta matriz se calculase solo cuando fuese necesaria.

3. Tiempos medidos

Todos los tiempos han sido recogidos ejecutando los algoritmos desarrollados en la máquina Hendrix.

En la siguiente tabla pueden verse los tiempos de ejecución obtenidos para cada uno de los tamaños probados, medidos en nanosegundos.

Tamaño	Fuerza Bruta	Algoritmo Voraz	Programación Dinámica	Ramificación y poda
4	96546	302377	432861	346091
11	4992103476	320170	686321714	40427158
12	54586581707	315887	1881248353	12621086
13	672067287909	324124	4930454438	24097028
14	8970103811234	322587	13270683814	35930945
15	-	368937	36460223685	380285598
16	-	437804	97965996531	171680601
20	-	431873	3482541718754	1006524010
21	-	394200	9422743308327	2437592003
26	-	502069	std::bad_alloc	std::bad_alloc
26	-	504694	std::bad_alloc	7582342908
27	-	553021	std::bad_alloc	32368051343
28	-	547310	std::bad_alloc	116559455193
29	-	718873	std::bad_alloc	40478898810
31	-	646382	std::bad_alloc	89883232394
36	-	742269	std::bad_alloc	131757576897
48	-	980611	std::bad_alloc	std::bad_alloc
100	-	3966709	-	-

Cuadro 1: Control de predicción

4. Opciones de compilación

El entorno empleado para la ejecución de los diferentes algoritmos es hendrix, conformado por 4 núcleos UltraSPARC-T2. Sabiendo esto, puede indicarse al compilador mediante ciertas opciones de compilación que aplique optimizaciones para el sistema. Estas son las siguientes:

- -O3: Opción genérica que indica al compilador que emplee todas las optimizaciones no específicas posibles.
- -Otime: Engloba O3 pero además habilita optimizaciones que pueden producir código que no sigue ciertos estándares (por ejemplo empleo de cálculos en coma flotante más rápido a costa de romper con varios estándares del IEEE).
- -mcpu=niagara2: Mediante esta opción especificamos al compilador exactamente que núcleo SPARC va a ejecutar el binario, por lo que aplicará optimizaciones propias para la CPU de hendrix, como instrucciones específicas de la arquitectura.
- -mfmaf: Indica al compilador que emplee instrucciones de multiplicación y suma flotante.

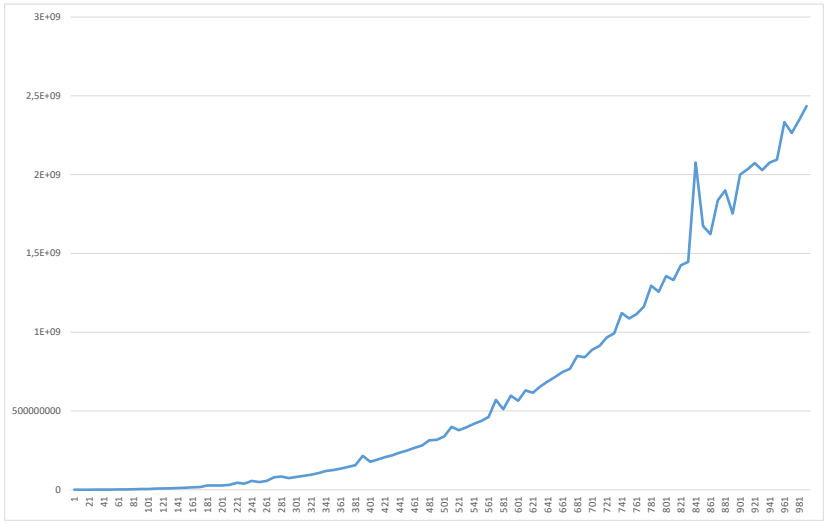

Figura 1: Gráfica AV

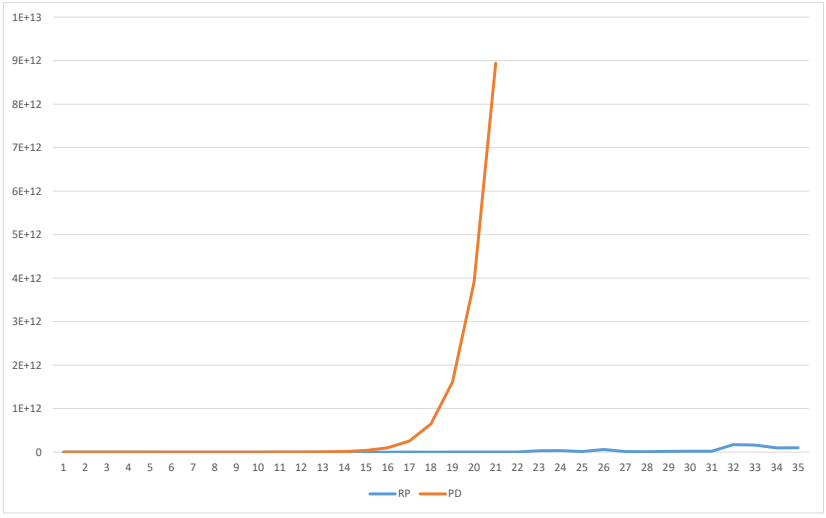

Figura 2: Gráfica RP Y PD

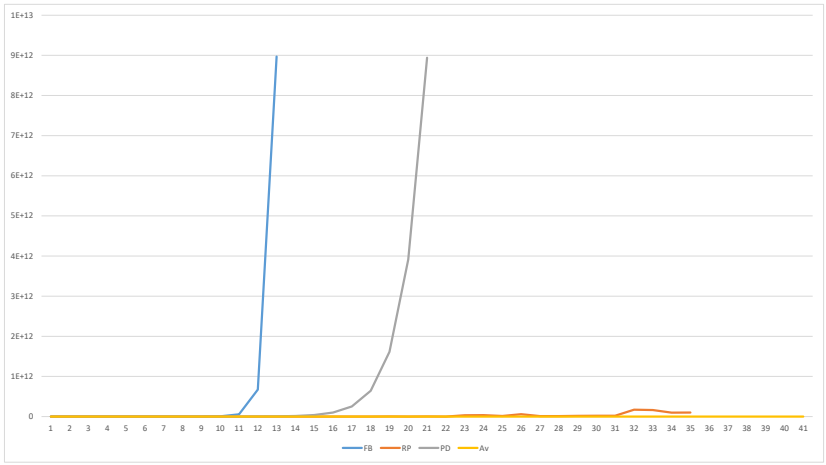

Figura 3: Comparación de todos los algoritmos