



INSTITUTO POLITÉCNICO NACIONAL



## **Análisis de Algoritmos**

---

*Reporte Programación Dinámica TSP*

Ingeniería en Sistemas Computacionales

Docente: Roberto Oswaldo Cruz Leija

Alumno: Héctor Mauricio Zamudio Domínguez

Fecha: 7 de noviembre de 2019

## Introducción

La programación dinámica puede ser una estrategia para resolver problemas para mejorar su optimización, reduciendo la complejidad computacional que conlleva resolver un problema además de el tiempo de solución. Sin embargo, es importante destacar en que situaciones es posible llevar esta estrategia y de qué forma se implementa.

## Fundamentos

La programación dinámica es un método para reducir el tiempo de ejecución de un algoritmo mediante la utilización de subproblemas superpuestos y subestructuras óptimas.

Es importe remarcar que la implementación de este tipo de programación es posible cuando de dichos subproblemas; los subproblemas son resueltos y esta solución ayuda a solución general del problema. Utilizar programación dinámica implica realizar una estructura que sea de utilizada para guardar y consultar las soluciones de los problemas de manera sencilla.

El Problema del Vendedor Viajero (conocido también como Travelling Salesman Problem o simplemente TSP) consiste en encontrar el circuito óptimo (en términos del viaje más corto) que deberá seguir un vendedor en un caso con  $n$  ciudades, en el que cada ciudad se visita exactamente una vez. Básicamente es una adaptación del Problema de Asignación que considera restricciones adicionales que garantiza la exclusión de subcircuitos en la solución óptima.

En teoría de grafos, este problema puede ser visto como encontrar un circuito Hamiltoniano, dependiendo del nodo donde se inicie el recorrido.

La programación dinámica es aplicable en este caso debido a que, posee subproblemas cuya solución aporta a la solución original. Por ejemplo, si encontramos la manera optima de recorrer 3 ciudades ya conocemos las rutas mínimas para volver a ciudad original y si se agrega una ciudad más, solo resta calcular cual es la vía mas corta para llegar a ella.

## Procedimiento

La estructura de datos que da sentido a la programación dinámica de esta solución es un bit estado, que representa las ciudades visitadas y las restantes por recorrer, asignando 1 para una ciudad recorrida y 0 a una ciudad no recorrida.

Creando un algoritmo que busca la ruta más corta hasta lograr completar este bit de estados completamente en 1, significando que recorrieron todas las ciudades volviendo a la ciudad original, al momento de encontrar esta ruta calcula las diferentes formas de recorrer las ciudades calculando su distancia y almacenando la menor(ruta) en el bit de estados. Como se menciono anteriormente esto da sentido a la parte dinámica ya que únicamente se realiza el proceso de calcular costes de recorrido para cada nueva ciudad ya que la mejor combinación fue calculada en iteraciones anteriores y únicamente es consultada esta información.

Implementando esta estrategia reduce de manera significativa la complejidad del problema, si se optara por una estrategia como la fuerza bruta se tendría que calcular todas las posibles permutaciones de formas de recorrer las ciudades, calcular el costo para cada ciclo y determinar el menor, dando una complejidad de orden factorial, lo cual aun que es totalmente eficaz, usar esta estrategia para problemas con mas de 20 ciudades ya presente problemas en tiempos de respuesta.

## Resultados y conclusiones

Ejemplo 1. Para la siguiente ciudad se calcula la ruta mínima.

	A	B	C	D
A	0	3	1	2
B	3	0	6	5
C	1	6	0	2
D	2	5	2	0

Resultado para ciudad inicial 0 (Ciudad A)

Tour: [0, 2, 3, 1, 0]  
Tour cost: 11.0

Resultado para ciudad inicial 3 (Ciudad D)

Tour: [3, 2, 0, 1, 3]  
Tour cost: 11.0

Ejemplo 2. Para la siguiente ciudad se calcula la ruta mínima

	a	b	c	d	e	f	g	h	i	j	
a	0	13	33	28	37	7	32	40	80	26	
b	13	0	39	83	50	68	16	98	81	55	
c	33	39	0	80	88	49	53	75	63	55	
d	28	83	80	0	94	4	20	6	59	76	
e	37	50	88	94	0	81	87	85	4	19	
f	7	68	49	4	81	0	96	53	40	37	
g	32	16	53	20	87	96	0	80	57	68	
h	40	98	75	6	85	53	80	0	65	41	
i	80	81	63	59	4	40	57	65	0	97	
j	26	55	55	76	19	37	68	41	97	0	

Resultado para ciudad inicial 0 (Ciudad A)

Tour: [0, 5, 8, 4, 9, 7, 3, 6, 1, 2, 0]  
Tour cost: 225.0

Resultado para ciudad inicial 8 (Ciudad i)

Tour: [8, 5, 0, 2, 1, 6, 3, 7, 9, 4, 8]  
Tour cost: 225.0

Puede observarse que para ambas pruebas existe un ciclo único que representa la ruta mínima de manera optima, el sentido en que se recorre dependerá de la ciudad inicial además de la ciudad adyacentes más próxima. Esto es debido a que las ciudades descritas son simétricas, es decir, tiene el mismo costo recorrer De una ciudad A hacia una ciudad B.

La programación vuelve a ser una herramienta útil para la resolución de problemas donde es posible subdividirse y resolver dichas subdivisiones(subproblemas) ayudan a la solución general del problema, la dificultad de utilizar programación dinámica radica en encontrar la estructura de datos que nos permita almacenar y consular las soluciones previamente calculadas y a partir de estas ir construyendo la solución general del problema.

Referencias

[https://es.wikipedia.org/wiki/Programaci%C3%B3n\\_din%C3%A1mica](https://es.wikipedia.org/wiki/Programaci%C3%B3n_din%C3%A1mica)

<https://www.gestiondeoperaciones.net/programacion-entera/solucion-del-problema-del-vendedor-viajero/#:~:targetText=El%20Problema%20del%20Vendedor%20Viajero,se%20visita%20exactamente%20una%20vez.>