

**Universidad Autónoma de Nuevo León**  
**Facultad de Ciencias Físico Matemáticas**  
**Matemáticas Computacionales**

Ismael Medina Robledo

1744617

**Reporte de algoritmo de Krustal**

**Resumen**

Implementación de un algoritmo de aproximación usando el algoritmo de Krustal para obtener un árbol de expansión mínima y de él crear una solución aproximada al problema del agente viajero.

**Problema del agente viajero**

El problema del agente viajero (TSP por sus siglas en inglés), responde a la siguiente pregunta: dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y al finalizar regresa a la ciudad origen?

Este tipo de problemas tiene gran aplicación en el ámbito de la logística y distribución, así como en la programación de curvas de producción.

**¿Qué es lo difícil del problema del agente viajero?**

La complejidad del cálculo del problema del agente viajero ha despertado múltiples iniciativas por mejorar la eficiencia en el cálculo de rutas.

El problema del agente viajero tiene una variación importante, y esta depende de que las distancias entre un nodo y otro sean simétricas o no, es decir, que la distancia entre A y B sea igual a la distancia entre B y A, puesto que en la práctica es muy poco probable que así sea.

La cantidad de rutas posibles en una red está determinada por la ecuación:

$(n-1)!$

## ¿Qué es un algoritmo de aproximación?

En situaciones donde todos los algoritmos conocidos son lentos, vale la pena considerar la posibilidad de usar una solución aproximada, o sea, una solución que tiene un valor de la función objetivo cerca del valor óptimo, pero no necesariamente el óptimo mismo.

Depende del área de aplicación si o no se puede hacer esto eficientemente. En muchos casos es posible llegar a una solución aproximada muy rápidamente mientras encontrar la solución óptima puede ser imposiblemente lento.

Un algoritmo de aproximación puede ser determinista o no determinista.

## ¿Qué hace el algoritmo de Kruskal?

El objetivo del algoritmo de Kruskal es construir un árbol (subgrafo sin ciclos) formado por arcos sucesivamente seleccionados de mínimo peso a partir de un grafo con pesos en los arcos.

## ¿Qué es un árbol de expansión mínima?

Dado un grafo conexo, no dirigido y con pesos en las aristas, un árbol de expansión mínima es un árbol compuesto por todos los vértices y cuya suma de sus aristas es la de menor peso.

## Descripción del ejemplo

El ejemplo que mostrare usara como datos las distancias en kilómetros de distintos municipios de Nuevo León.

## Soluciones

1. El método de la fuerza bruta no implica la aplicación de ningún algoritmo sistemático, tan solo consiste en explorar todos los recorridos posibles. Considerando la siguiente red simétrica, los caminos posibles se reducen a la mitad.
2. El método del vecino más cercano es un algoritmo heurístico diseñado para solucionar el problema del agente viajero, no asegura una solución óptima, sin embargo, suele proporcionar buenas soluciones, y tiene un tiempo de cálculo muy eficiente. El método de desarrollo es muy similar al utilizado para resolver problemas de árbol de expansión mínima.
3. El método de branch and bound (ramificación y poda) nos proporciona una solución óptima del problema del agente viajero, calculando mediante el algoritmo simplex la solución del modelo. A medida que aumente el tamaño de la red el método puede tardar gran cantidad de tiempo en resolverse, sin embargo, para redes de mediano tamaño es una excelente alternativa.

## Heurística del vecino más cercano

El método del vecino más cercano es un algoritmo heurístico diseñado para solucionar el problema del agente viajero, no asegura una solución óptima, sin embargo, suele proporcionar buenas soluciones, y tiene un tiempo de cálculo muy eficiente. El método de desarrollo es muy similar al utilizado para resolver problemas de árbol de expansión mínima.

## Solución exacta al problema

La solución más directa puede ser, intentar todas las permutaciones (combinaciones ordenadas) y ver cuál de estas es la menor (usando una Búsqueda de fuerza bruta). El tiempo de ejecución es un factor polinómico de

orden  $O(n!)$ , el Factorial del número de ciudades, esta solución es impracticable para dado solamente 20 ciudades. Una de las mejores aplicaciones de la Programación dinámica es el algoritmo Held–Karp que resuelve el problema en  $O(n^2 2^n)$ .

Para el ejemplo se usaron los siguientes datos:

Desde	Hasta	Distancia (km)
('Apodaca', 'China', 95)		
('Apodaca', 'Arramberri', 190)		
('Apodaca', 'Doctor Arroyo', 234)		
('Apodaca', 'Monterrey', 16)		
('Apodaca', 'Cadereyta', 28)		
('Apodaca', 'Pesqueria', 13)		
('Apodaca', 'San Nicolas', 10)		
('Apodaca', 'San Pedro', 25)		
('Apodaca', 'Lampazos', 141)		
('China', 'Arramberri', 187)		
('China', 'Doctor Arroyo', 245)		
('China', 'Monterrey', 108)		
('China', 'Cadereyta', 77)		
('China', 'Pesqueria', 81)		
('China', 'San Nicolas', 105)		
('China', 'San Pedro', 116)		
('China', 'Lampazos', 193)		
('Arramberri', 'Doctor Arroyo', 60 )		
('Arramberri', 'Monterrey', 183)		
('Arramberri', 'Cadereyta', 166)		
('Arramberri', 'Pesqueria', 189)		
('Arramberri', 'San Nicolas', 189)		
('Arramberri', 'San Pedro', 183)		
('Arramberri', 'Lampazos', 332)		
('Doctor Arroyo', 'Monterrey', 224)		
('Doctor Arroyo', 'Cadereyta', 213)		
('Doctor Arroyo', 'Pesqueria', 235)		
('Doctor Arroyo', 'San Nicolas', 231)		
('Doctor Arroyo', 'San Pedro', 231)		
('Doctor Arroyo', 'Lampazos', 374)		
('Monterrey', 'Cadereyta', 33)		
('Monterrey', 'Pesqueria', 28 )		
('Monterrey', 'San Nicolas', 7)		
('Monterrey', 'San Pedro', 9)		
('Monterrey', 'Lampazos', 149)		
('Cadereyta', 'Pesqueria', 22)		
('Cadereyta', 'San Nicolas', 34)		
('Cadereyta', 'San Pedro', 41)		
('Cadereyta', 'Lampazos', 167)		
('Pesqueria', 'San Nicolas', 24)		
('Pesqueria', 'San Pedro', 37)		
('Pesqueria', 'Lampazos', 145 )		
('San Nicolas', 'San Pedro', 15)		
('San Nicolas', 'Lampazos', 143)		
('San Pedro', 'Lampazos', 152)		

Usando el algoritmo de Krustal obtuvimos que la mejor ruta es la siguiente:

Desde	Hasta	
Apodaca	Pesqueria	13
Pesqueria	Cadereyta	22
Cadereyta	China	77
China	Arramberri	187
Arramberri	Doctor Arroyo	60
Doctor Arroyo	San Nicolas	231
San Nicolas	Monterrey	7
Monterrey	San Pedro	9
San Pedro	Lampazos	152
Lampazos	Apodaca	141
Costo	899	

## Conclusiones

Con este algoritmo podemos encontrar una muy buena solución en un muy poco tiempo, y aunque no es la mejor el poco tiempo que tarda en encontrar la solución hace que sea la mejor ruta a escoger ya que es una de las mejores y la obtienes en poco tiempo. En cambio, si quieres la mejor de todas las soluciones tendrás que esperar demasiado tiempo para obtenerla.