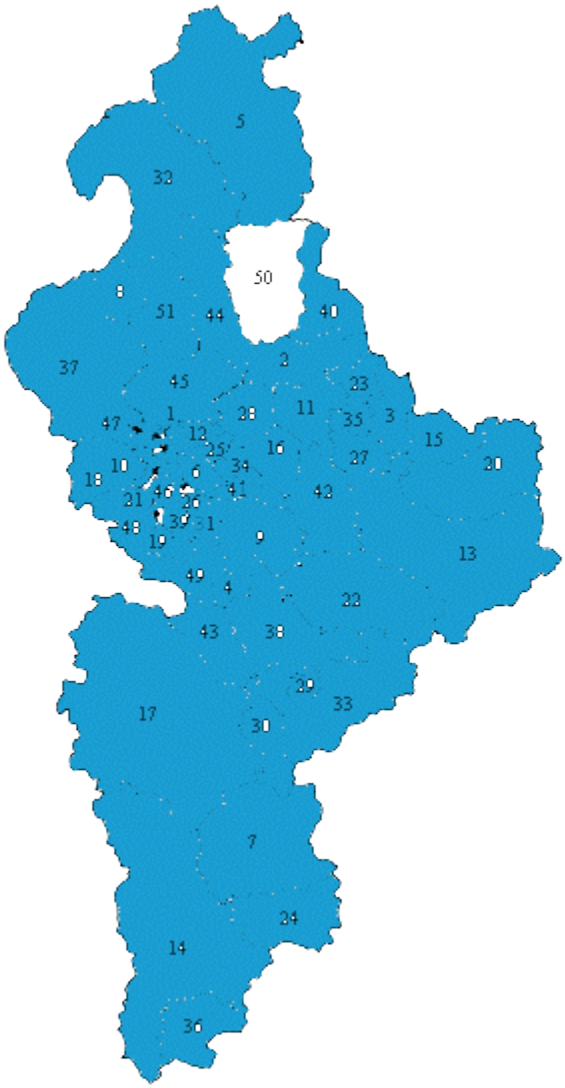


Problema del viajero Nuevo Leones

PROYECTO FINAL

Ismael Medina | | 26 de noviembre de 2017

Abstract.

¿Cuál es la ruta más corta posible que visita 50 municipios del estado de Nuevo León exactamente una vez y al finalizar regresa a la ciudad origen? Esta es la pregunta que responde la implementación del algoritmo Kruskal en nuestro grafo de cincuenta nodos y mil doscientos veinte y cinco aristas, en el cual también fue implementado el algoritmo de "el vecino más cercano". Dando así el código del algoritmo en Python.

Introducción.

El problema del viajero consta (o se puede resumir) de dos partes, tener lugares que recorrer y minimizar gastos. Para la primera parte, se introduce un concepto matemático que es objeto de estudio en la Teoría de grafos, un grafo conexo ponderado, y para la segunda parte se hace uso de un algoritmo muy conocido por los informáticos, el algoritmo Kruskal. Es decir, este problema cae en el área de las matemáticas aplicadas.

En este proyecto lo que nos interesa recorrer es el estado de Nuevo León, un viaje por 50 de sus municipios y para esto, no solo usaremos el algoritmo Kruskal si no, otro algoritmo también muy conocido llamado Vecino más cercano.

Para este proyecto es necesario que se conozcan los conceptos de *grafo*, *grafo conexo* y *grafo ponderado*, conceptos matemáticos que tienen una aplicación en ciencias de la computación. Además de esto, saber que es un *algoritmo* y que función tienen el *algoritmo kruskal* y "*vecino más cercano*" en relación con un grafo.

Dato curioso. El primer artículo científico relativo a grafos fue escrito por el matemático suizo Leonhard Euler en 1736.

Problema del Agente Viajero

El objetivo es encontrar un recorrido completo que conecte todos los nodos de una red, visitándolos tan solo una vez y volviendo al punto de partida, y que además minimice la distancia total de la ruta. El número de posibles soluciones es tan elevado que si pretendemos que el algoritmo encargado de la búsqueda de la solución óptima deba verificar una a una no tendremos tiempo de cálculo para hallarlo: es probable que en el peor de los casos el tiempo de resolución de cualquier algoritmo para PAV aumente exponencialmente con el número de ciudades, por lo que incluso en algunos casos de tan sólo cientos de ciudades se tardarán bastantes años de CPU para resolverlos de manera exacta.

¿Qué es lo “difícil” en el Problema del Agente Viajero?

En el ámbito de la teoría de complejidad computacional, el PAV pertenece a la clase de problemas NP-completos. Por lo tanto, se supone que no hay ningún algoritmo eficiente para la solución de PAV. En otras palabras, el número de posibles soluciones es tan elevado que si pretendemos que el algoritmo encargado de la búsqueda de la solución óptima deba verificar una a una no tendremos tiempo de cálculo para hallarlo: es probable que en el peor de los casos el tiempo de resolución de cualquier algoritmo para PAV aumente exponencialmente con el número de ciudades, por lo que incluso en algunos casos de tan sólo cientos de ciudades se tardarán bastantes años de CPU para resolverlos de manera exacta.

¿Qué es un algoritmo de aproximación?

Un algoritmo de aproximación es un algoritmo usado para encontrar soluciones aproximadas a problemas de optimización. A diferencia de las heurísticas, que usualmente sólo encuentran soluciones razonablemente buenas en tiempos razonablemente rápidos, lo que se busca aquí es encontrar soluciones que está demostrado son de calidad y cuyos tiempos de ejecución están acotadas por cotas conocidas. Idealmente, la aproximación mejora su calidad para factores constantes pequeños.

Definiciones para una mejor comprensión.

Def. Grafo. Es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten relaciones binarias entre elementos de un conjunto.

Def. Grafo conexo. Un grafo se dice conexo si, para cualquier par de vértices u y v en G, existe al menos una trayectoria de u a v.

Def. Grafo ponderado. Grafo al cual se le ha añadido un peso a las aristas. Generalmente suele ser un número natural.

Def. Algoritmo. Conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite llevar a cabo una actividad mediante pasos sucesivos.

Muchos algoritmos son ideados para implementarse en un programa.

Def.Árbol:Un árbol es una gráfica en la cual no existen ciclos.

Def.Árbol de expansión: Un árbol de expansión es aquel árbol que enlaza todos los nodos de la red, de igual manera no permite la existencia de ciclos.

Def. Algoritmo Kruskal. Algoritmo para encontrar un árbol de expansión mínima en un grafo ponderado.

Def. Algoritmo Vecino más cercano. También llamado algoritmo voraz (greedy) permite al viajante elegir la ciudad no visitada más cercana como próximo movimiento. Este algoritmo retorna rápidamente una ruta corta.

Esto es,

- Se toma un vértice origen y se agrega a una lista.
- Se revisa cuáles son sus vecinos, elige el de menor peso revisando que no esté en la lista. Se agrega a la lista.
- Así sucesivamente hasta que se tenga en la lista todos los vértices del grafo.
- Se agrega el vértice origen pero al final de la lista.

Para N ciudades aleatoriamente distribuidas en un plano, el algoritmo en promedio retorna un camino de un 25% más largo que el menor camino posible

Implementación en Python de los Algoritmos Kruskal y Heurística del Vecino más cercano.

Kruskal:

```
107
108     def shortest(self, v): # Dijkstra's algorithm
109         q = [(0, v, ())] # arreglo "q" de las "Tuplas" de lo que se va a almacenar dondo 0 es la distancia
110         dist = dict() #diccionario de distancias
111         visited = set() #Conjunto de visitados
112         while len(q) > 0: #mientras exista un nodo pendiente
113             (l, u, p) = heappop(q) # Se toma la tupla con la distancia menor
114             if u not in visited: # si no lo hemos visitado
115                 visited.add(u) #se agrega a visitados
116                 dist[u] = (l,u,list(flatten(p))[::-1] + [u])    #agrega al diccionario
117             p = (u, p) #Tupla del nodo y el camino
118             for n in self.vecinos[u]: #Para cada hijo del nodo actual
119                 if n not in visited: #si no lo hemos visitado
120                     el = self.E[(u,n)] #se toma la distancia del nodo acutal hacia el nodo hijo
121                     heappush(q, (l + el, n, p)) #Se agrega al arreglo "q" la distancia actual mas la ditan
122         return dist #regresa el diccionario de distancias
123
124     def kruskal(self):
125         e = deepcopy(self.E)
126         arbol = Grafo()
127         peso = 0
128         comp = dict()
129         t = sorted(e.keys(), key = lambda k: e[k], reverse=True)
130         nuevo = set()
131         while len(t) > 0 and len(nuevo) < len(self.V):
132             #print(len(t))
133             arista = t.pop()
134             w = e[arista]
135             del e[arista]
136             (u,v) = arista
137             c = comp.get(v, {v})
138             if u not in c:
139                 #print('u ',u, 'v ',v ,'c ', c)
140                 arbol.conecta(u,v,w)
141                 peso += w
142                 nuevo = c.union(comp.get(u,{u}))
143                 for i in nuevo:
144                     comp[i]= nuevo
145         print('MST con peso', peso, ':', nuevo, '\n', arbol.E)
146         return arbol
147
```

Vecino más cercano:

```
147
148     def vecinoMasCercano(self):
149         ni = random.choice(list(self.V))
150         result=[ni]
151         while len(result) < len(self.V):
152             ln = set(self.vecinos[ni])
153             le = dict()
154             res =(ln-set(result))
155             for nv in res:
156                 le[nv]=self.E[(ni,nv)]
157             menor = min(le, key=le.get)
158             result.append(menor)
159             ni=menor
160         return result
161
```

Descripción del Problema del viajero Nuevo Leones.

Para una representación del algortimo, se hará uso de todo el Estado de Nuevo León, exceptuando el municipio de Vallecilo, cuyo listado de los municipios es el siguiente:

1. Abasolo
2. Agualeguas
3. Allende
4. Anáhuac
5. Apodaca
6. Aramberri
7. Bustamante
8. Cadereyta Jiménez
9. Cerralvo
10 . China
11. Ciénega de Flores
12. Dr. Arroyo
13. Dr. Coss
14. Dr. González
15. El Carmen
16. Galeana
17. García
18. Gral. Bravo
19. Gral. Escobedo
20. Gral. Terán
21. Gral. Treviño
22. Gral. Zaragoza
23. Gral. Zuazua
24. Guadalupe
25. Hida.Hidalgo
26. Higueras
27. Hualahuises
28. Iturbide
29. Juárez
30. Lampazos de Naranjo
31. Linares
32. Los Aldamas
33. Los Herreras
34. Los Ramones
35. Marín
36. Melchor Ocampo
37. Mier y Noriega
39. Montemorelos
40. Monterrey
41. Parás
42. Pesquería
43. Rayones
44. Sabinas Hidalgo
45. Salinas Victoria
46. San Nicolás de los Garza
47. San Pedro Garza García
48. Santa Catarina
49. Santiago
50. Villaldama

Grafo de los 50 municipios seleccionados del estado de Nuevo León:
https://github.com/PaulinaAguirre/1837503MC/blob/master/Proyecto/Grafo

Tomando una muestra de los 100 resultados obtuvimos lo siguiente:

En Kruskal:

Para no poner todos los municipios seleccionados en esta muestra, solo se pondrá la mejor solución de ésta.

```
Camino iniciando en  Villa
De  Villa       a Bust  la distancia es  20.396078054371138
De  Bust        a Mina  la distancia es  26.076809620810597
De  Mina        a Hida  la distancia es  36.71511950137164
De  Hida        a Aba   la distancia es  4.47213595499958
De  Aba         a Carmen        la distancia es  8.54400374531753
De  Carmen      a Esco  la distancia es  4.47213595499958
De  Esco        a Pedro         la distancia es  9.055385138137417
De  Pedro       a Nico  la distancia es  12.083045973594572
De  Nico        a Guada         la distancia es  5.385164807134504
De  Guada       a Mty   la distancia es  9.433981132056603
De  Mty         a Santiago      la distancia es  11.661903789690601
De  Santiago    a Ray   la distancia es  18.681541692269406
De  Ray         a Gale  la distancia es  35.510561809129406
De  Gale        a All   la distancia es  53.600373133029585
De  All         a Monte         la distancia es  18.867962264113206
De  Monte       a Huala         la distancia es  24.698178070456937
De  Huala       a Linares       la distancia es  20.09975124224178
De  Linares     a Terán         la distancia es  24.186773244895647
De  Terán       a Itur  la distancia es  55.60575509783138
De  Itur        a Aram  la distancia es  23.021728866442675
De  Aram        a Zara  la distancia es  22.825424421026653
De  Zara        a MN    la distancia es  43.382023926967726
De  MN  a Apo   la distancia es  145.08618128546908
De  Apo         a Zua   la distancia es  9.219544457292887
De  Zua         a Marín         la distancia es  8.246211251235321
De  Marín       a Hig   la distancia es  11.704699910719626
De  Hig         a Pesque        la distancia es  18.027756377319946
De  Pesque      a Cade  la distancia es  16.401219466856727
De  Cade        a DG    la distancia es  17.46424919657298
De  DG  a Cerra         la distancia es  15.264337522473747
De  Cerra       a MC    la distancia es  17.11724276862369
De  MC  a Trevi         la distancia es  14.866068747318506
De  Trevi       a Parás         la distancia es  23.259406699226016
De  Parás       a Herreras      la distancia es  41.43669871020132
De  Herreras    a Ramones       la distancia es  17.029386365926403
De  Ramones     a Aldamas       la distancia es  34.20526275297414
De  Aldamas     a Coss  la distancia es  24.515301344262525
De  Coss        a Bravo         la distancia es  19.924858845171276
De  Bravo       a China         la distancia es  24.166091947189145
De  China       a Agua  la distancia es  95.29428104561154
De  Agua        a Cién  la distancia es  44.28317965096906
De  Cién        a Juarez        la distancia es  20.8806130178211
De  Juarez      a Santa         la distancia es  36.013886210738214
De  Santa       a Garcia        la distancia es  20.615528128088304
De  Garcia      a Naranjo       la distancia es  75.39230729988306
De  Naranjo     a Arr   la distancia es  18.601075237738275
De  Arr         a Aná   la distancia es  19.235384061671343
De  Aná         a Sabinas       la distancia es  45.70557952810576
De  Sabinas     a  Villa        la distancia es  20.615528128088304
Suma del camino:  1343.3477173984363
```

Y por lo tanto el mejor camino para esta muestra fue el siguiente:

```
El mejor camino fue el siguiente:
Villa ->
Bust ->
Mina ->
Hida ->
Aba ->
Carmen ->
Esco ->
Pedro ->
Nico ->
Guada ->
Mty ->
Santiago ->
Ray ->
Gale ->
All ->
Monte ->
Huala ->
Linares ->
Terán ->
Itur ->
Aram ->
Zara ->
MN ->
Apo ->
Zua ->
Marín ->
Hig ->
Pesque ->
Cade ->
DG ->
Cerra ->
MC ->
Trevi ->
Parás ->
Herreras ->
Ramones ->
Aldamas ->
Coss ->
Bravo ->
China ->
Agua ->
Cién ->
Juarez ->
Santa ->
Garcia ->
Naranjo ->
Arr ->
Aná ->
Sabinas ->
Villa

Con un costo de  1343.3477173984363
Tiempo de ejecucion:  24.448792304608546
```

En Heurística del Vecino Más Cercano:

Para no poner todos los municipios seleccionados en esta muestra, solo se pondrá la mejor solución de ésta.

```
Camino iniciando en  Santa
De  Santa       a Pedro         la distancia es  13.341664064126334
De  Pedro       a Esco  la distancia es  9.055385138137417
De  Esco        a Carmen        la distancia es  4.47213595499958
De  Carmen      a Aba   la distancia es  8.54400374531753
De  Aba         a Hida  la distancia es  4.47213595499958
De  Hida        a Garcia        la distancia es  22.360679774997898
De  Garcia      a Mina  la distancia es  30.463092423455635
De  Mina        a Bust  la distancia es  26.076809620810597
De  Bust        a Villa         la distancia es  20.396078054371138
De  Villa       a Sabinas       la distancia es  20.615528128088304
De  Sabinas     a Hig   la distancia es  33.54101966249684
De  Hig         a Marín         la distancia es  11.704699910719626
De  Marín       a Zua   la distancia es  8.246211251235321
De  Zua         a Cién  la distancia es  8.06225774829855
De  Cién        a Apo   la distancia es  10.0
De  Apo         a Guada         la distancia es  5.0990195135927845
De  Guada       a Nico  la distancia es  5.385164807134504
De  Nico        a Mty   la distancia es  10.0
De  Mty         a Santiago      la distancia es  11.661903789690601
De  Santiago    a Juarez        la distancia es  14.317821063276353
De  Juarez      a Pesque        la distancia es  15.652475842498529
De  Pesque      a Cade  la distancia es  16.401219466856727
De  Cade        a DG    la distancia es  17.46424919657298
De  DG  a Cerra         la distancia es  15.264337522473747
De  Cerra       a Agua  la distancia es  13.0
De  Agua        a MC    la distancia es  26.076809620810597
De  MC  a Trevi         la distancia es  14.866068747318506
De  Trevi       a Herreras      la distancia es  18.439088914585774
De  Herreras    a Ramones       la distancia es  17.029386365926403
De  Ramones     a Terán         la distancia es  28.653097563788805
De  Terán       a Linares       la distancia es  24.186773244895647
De  Linares     a Huala         la distancia es  20.09975124224178
De  Huala       a Itur  la distancia es  24.186773244895647
De  Itur        a Aram  la distancia es  23.021728866442675
De  Aram        a Zara  la distancia es  22.825424421026653
De  Zara        a MN    la distancia es  43.382023926967726
De  MN  a Gale  la distancia es  90.13878188659973
De  Gale        a Ray   la distancia es  35.510561809129406
De  Ray         a All   la distancia es  18.384776310850235
De  All         a Monte         la distancia es  18.867962264113206
De  Monte       a Aldamas       la distancia es  79.20858539325141
De  Aldamas     a Coss  la distancia es  24.515301344262525
De  Coss        a Bravo         la distancia es  19.924858845171276
De  Bravo       a China         la distancia es  24.166091947189145
De  China       a Parás         la distancia es  86.97700845625813
De  Parás       a Aná   la distancia es  67.35725647619564
De  Aná         a Arr   la distancia es  19.235384061671343
De  Arr         a Naranjo       la distancia es  18.601075237738275
De  Naranjo     a  Santa        la distancia es  83.86298349093002
Suma del camino:  1205.1154463164112
```

Y por lo tanto el mejor camino para esta muestra fue el siguiente:

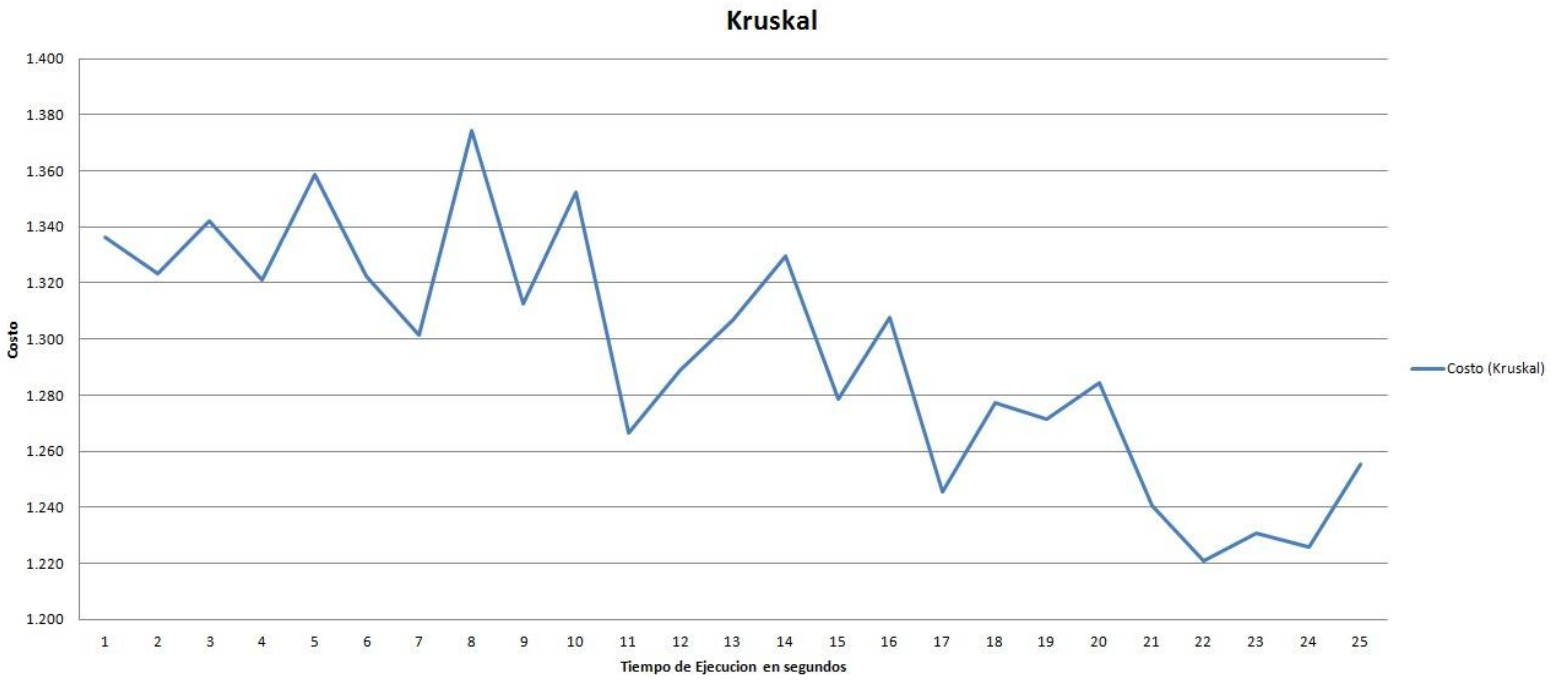
```
El mejor camino fue el siguiente:
Santa ->
Pedro ->
Esco ->
Carmen ->
Aba ->
Hida ->
Garcia ->
Mina ->
Bust ->
Villa ->
Sabinas ->
Hig ->
Marín ->
Zua ->
Cién ->
Apo ->
Guada ->
Nico ->
Mty ->
Santiago ->
Juarez ->
Pesque ->
Cade ->
DG ->
Cerra ->
Agua ->
MC ->
Trevi ->
Herreras ->
Ramones ->
Terán ->
Linares ->
Huala ->
Itur ->
Aram ->
Zara ->
MN ->
Gale ->
Ray ->
All ->
Monte ->
Aldamas ->
Coss ->
Bravo ->
China ->
Parás ->
Aná ->
Arr ->
Naranjo ->
Santa

Con un costo de  1205.1154463164112
Tiempo de ejecucion:  20.121760377366392
```

Resultados.

Al correr el algoritmo de Kruskal, para 1, 2, 3, 4 y 5 a la vez, se obtuvo lo siguiente:

Costo (Kruskal)	Tiempo de ejecucion
1.336	4
1.323	5
1.342	5
1.321	5
1.359	5
1.322	11
1.301	4,614880697
1374,326301	12
1.312	12,43095247
1.352	13
1266,722214	14
1289,087919	14,89325682
1306,632793	15,04399889
1329,555657	15
1.279	15,69292562
1307,860339	15,4115903
1245,740593	16
1277,448686	17
1271,452951	16,93546379
1284,312329	17,26813024
1240,547813	18,62541237
1221,159752	18,71496038
1230,960639	18,67018637
1226,060196	18,72125193
1255,53	18,72

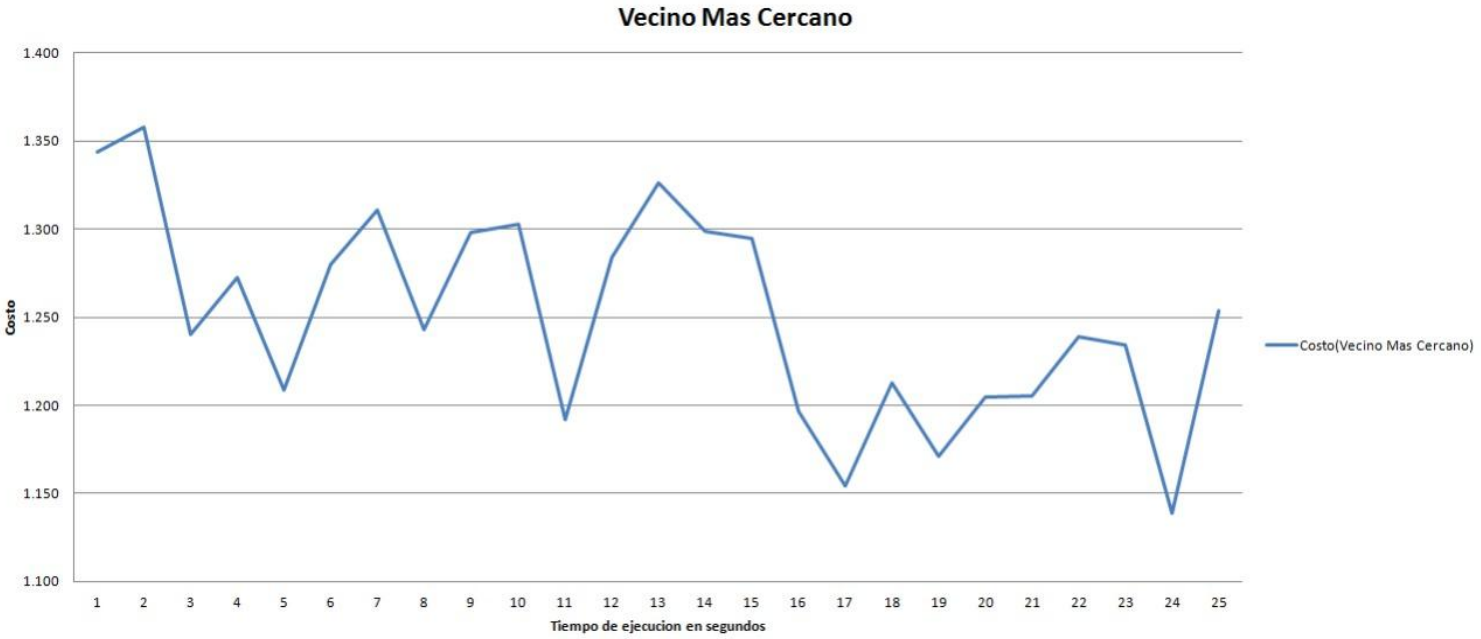
Gráfica Kruskal:



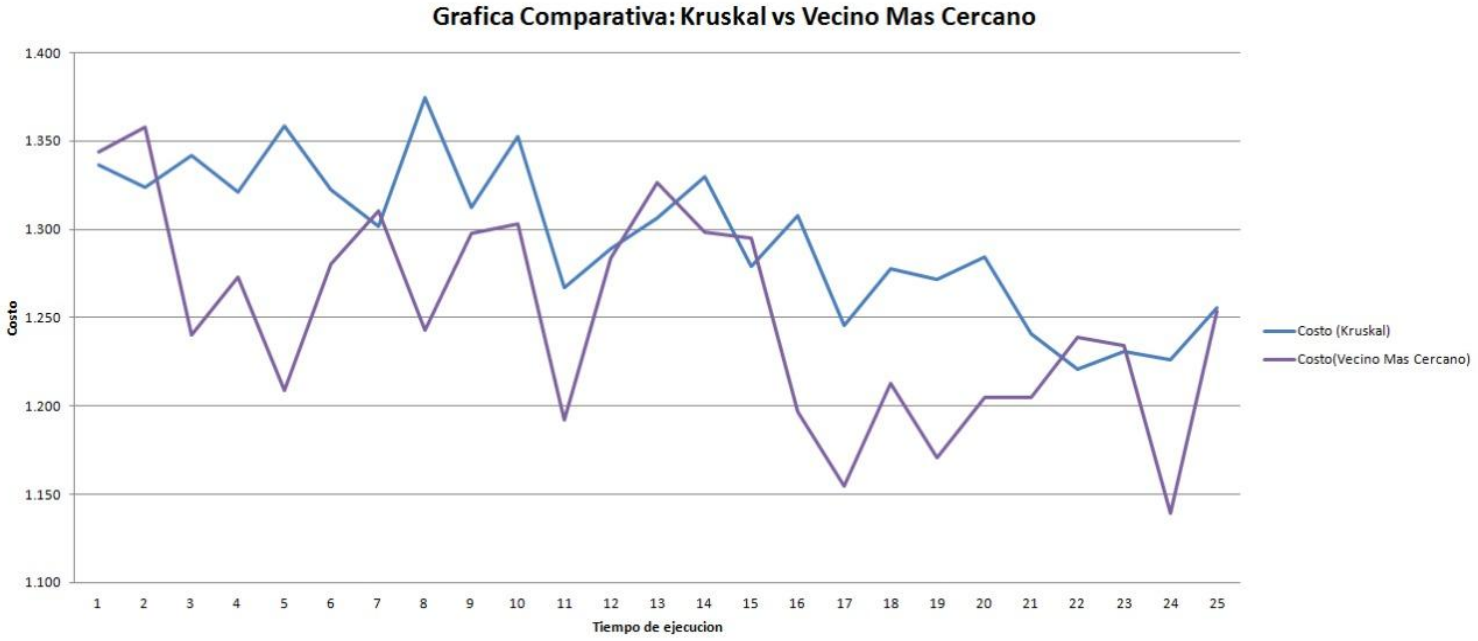
Al correr el algoritmo de Kruskal, para 1, 2, 3, 4 y 5 a la vez, se obtuvo lo siguiente:

Costo(Vecino Mas Cercano)	Tiempo de ejecucion
1.343	4
1.358	4
1.240	5
1.273	5
1.209	4
1280,04894	12
1.311	9
1.243	11
1.298	10
1.303	9
1191,774666	16,90452299
1283,81021	15,1003409
1326,632758	16,78426519
1298,474129	14,7961582
1294,718964	16,79461382
1196,762222	17
1154,469479	15
1212,961215	17
1170,796126	18
1204,793687	17
1.205	25,36
1239,02149	20,27782253
1234,250258	21,4791254
1139,145871	20,14820367
1253,712548	24,71290163

Grafica Vecino Más Cercano:



Gráfica Comparativa entre Kruskal y Heurística del Vecino Más Cercano:



Conclusión

Pudimos observar que el problema del agente viajero, no es muy simple que digamos, ya que no existe un algoritmo que determine la ruta más eficiente por una cantidad significativa de tiempo. Es decir, para tener la solución más óptima, se necesita correr todas las combinaciones entre ciudades.

Por otro lado existen alternativas de algoritmos, que a pesar de no ser exactos, se pueden usar, para que en cierta cantidad de veces que se corra se encuentre una solución “buena” para el problema, ya sea el algoritmo de kruskal, o el del vecino más cercano.

Se pudo observar que mientras se incrementaba el numero de ciudades de manera aleatoria, el costo de la distancia era no el mejor de todos, pero si eficiente, para determina una distancia corta, la desventaja era que mientras más ciudades se tomaban, el tiempo de ejecución incrementaba, lo cual hace que a cierto punto no sea tan eficiente estos algoritmos.

Pudimos notar que a pesar de que existe una pequeña diferencia entre ambos algoritmos (Kruskal, usa pesos mínimos y árboles de expansión; mientras que VMC usa una ciudad fija elegida al azar), los resultados eran muy similares, unos mejores que otros por la aleatoriedad, pero muy cercanos entre ellos. En sí es difícil seleccionar uno mejor.

En cuanto a no usar el algoritmo con la solución exacta, estos dos algoritmos ya mencionados, sirven de alternativa a hallar una solución o recorrido “bueno”, ya que si no fuera así, este tardaría una gran cantidad de tiempo para poder obtener una solución exacta.

En sí para 50 ciudades no hubiera sido un problema, pero ¿para 1000 o 10000?, exacto sería mejor Kruskal o VMC.

Una pequeña desventaja al hacer uso de el algoritmo fue la implementación de los datos de las distancias entre todas las ciudades ya que si, con 50 fue mucho buscar, en total 1225 datos recolectados, para más serían aún más (se obtuvo con la siguiente ecuación $\sum_{i=1}^{n}(i-1)$, donde n es el total de ciudades en este caso 50)