El problema del camino mas corto sobre la red del metrobús y del metro de la Ciudad de México (Solución por Dijkstra y Simplex Dual)

Jesús Manuel Mager Hois

***Resumen*—El problema del camino mas corto desde un único nodo origen puede ser resuelto por diversos algoritmos. En este trabajo comparo la solución del problema mediante simplex dual y dijkstra, aplicado a la red del metro y del metrobús. Se plantean las complejidades de los dos algoritmos, así como datos estadísticos del comportamiento de los mismos.**

Keywords. Camino más corto, Metro, Metrobús, Ciudad de México, Dijkstra, Simplex.

I. Introducción

Dado un digrafo $G = (V, E)$, donde V es el número de vértices y E el número de aristas, trataremos de encontrar un camino $P = (v_{origen}, \dots, v_{destino})$, donde $v_i \in V$. Cada arista $(v_i, v_j) \in E$ tiene un costo de tránsito $c \in C$, donde C es el conjunto de todos los costos y c_i es el costo de la arista (v_i, v_j). De esta manera, un camino tendrá un costo de la suma de los costos correspondientes a todas las aristas transitadas. El camino con el costo mínimo será el camino óptimo o más corto.

El problema del actual trabajo es encontrar el camino P más corto, entre el nodo de origen v_{origen} y $v_{destino}$, con el menor tiempo ejecución y la menor complejidad posible. Para ello evaluaremos dos algoritmos que resuelven el problema, dijkstra [1] y simplex [2], el primero especializado en la búsqueda de mejores caminos y el segundo el algoritmo general para la resolución de problemas de programación lineal. Resolver el camino mas corto puede llegar a ser costoso en tiempo de ejecución. Existen diversos algoritmos que tratan el problema, pero la complejidad varía y en la práctica la los tiempos son muy diferentes. Por lo tanto es importante tener claro estas diferencias.

Por lo tanto nuestro objetivo es comparar dijkstra y simplex en el caso particular de encontrar el camino mas corto entre la estación Cuatro Caminos y Tlahuac, sobre la red del metro y metrobús de la ciudad de México. Trabajos parecidos han sido realizados para el metrobús en México [3] y en otras ciudades importantes [4]. Los datos para construir la res del metro y metrobús se obtuvieron de las páginas oficiales [5] [6].

II. Los algoritmos

II-A. Simplex

El algoritmo simplex es usado para resolver cualquier problema de programación lineal, por lo que no se reduce a la aplicación a la búsqueda de un camino óptimo. Sin embargo, es posible utilizarlo, con un planteamiento del problema particular. Las estaciones de las dos redes se tomarán como vértices y las conexiones entre ellos como aristas. La función a minimizar se compone de la suma de las aristas junto con su peso y las restricciones tendrán valores donde un nodo salida y objetivo serán iguales a uno y todos los demás se igualarán a cero. Todas las aristas serán positivias. Chandrasekaran [7] plantea el problema de la sigueinte forma:

$$\min \sum_{ij \in A} d_{ij} f_{ij}$$

Sujeto a:

$$\sum_j (f_{ij} - f_{ji}) = \begin{cases} \text{1 si i = s} \\ \text{-1 si i = t} \\ \text{0 si i} \neq s, t \end{cases}$$

$$f_{i,j} \geq 0, \forall (i,j) \in A$$

El problema de programación lineal dual es:

$$\max p_s - p_t$$
$$p_i - p_j \leq d_{i,j} \forall (i,j) \in A$$

Y sea $\pi_i = -p_i$ obtenemos las relaciones:

$$\min \pi_t - \pi_s$$
$$\pi_j - \pi_i \le d_{i,j} \forall (i,j) \in A$$

Si ambos problemas tienen una solución óptima, entonces $(\pi_i - \pi_s)$ tiene una interpretación de la distancia más corta de s a i. El problema dual es factible si no existe un ciclo dirigido $C \ni \sum_{(i,j)\in C} d_{i,j} < 0$. Si existe un ciclo así, el problema sería no acotado.

Como el planteamiento del modelo no corresponde a la forma canónica ya que las restricciones utilizan igualdades [2]:

$$\min c'x$$
$$Ax = b$$
$$x \ge 0$$

Será necesario utilizar un método de resolución adicional, como puede ser el método simplex de dos faces o el método simplex dual. En el caso particular de este trabajo se utilizará el dual.

II-B. Dijkstra

El algoritmo Dijkstra es el algoritmo que resuelve el problema de la búsqueda del mejor camino de con un único nodo de partida hacia todos los nodos del gráfico, y es hasta el momento el algoritmo más veloz, para este caso con optimizaciones posteriores. En 1959 Dijkstra [1] escribe su algoritmo en la revista "Numerische Mathematik"que tendrá un rendimiento, según sus análisis, de $O(n^2)$ en el peor caso donde n son el número de nodos del gráfico [8]. El algoritmo tiene también el supuesto de que el gráfico sobre el que se va a trabajar es de peso y el valor de cada uno de sus arcos debe ser positivo, o mayor que cero, de lo contrario se devolverá resultados erróneos o incluso pueden generarse ciclos infinitos [9]. Para gráficos que requieren utilizar arcos negativos se puede usar algoritmo *Bellman-Ford* [10] [11].

Algoritmo 1 Dijkstra [12]

```
Entrada: G, n_s
Salida: Distancia[], Memoria[]
dijkstra(G, n_s)
  ColaDePrioridad P ← {}
  Distancia[n_s] ← 0
  Para todo n en N:
    si n ≠ n_s entonces:
      Distancia[n] ← ∞
      Memoria[n] ← null
    fin si
    agregar n a P
  fin para todo
  Mientras S no este vacío:
    n ← min(P)
    eliminar n de P
    Para todo n_i ∈ Γ(n):
      c ← Distancia[n] + costo(n, n_i)
      si c < Distancia[n_i]
        Distancia[n_i] ← c
        Memoria[n_i] ← n
      Fin si
    Fin para todo
  Fin mientras
  Devuelve Distancia[], Memoria[]
Fin Dijkstra
```

III. Complejidad

Si bien se ha dicho, según el planteamiento original de Dijkstra su complejidad era del $O(n^2)$, en estudios posteriores se comenzaron a usar listas de prioridad para mejorar el rendimiento del algoritmo con *binary heaps*(montículo binarios) y *Fibonacci heaps* [13][1]. Con este aporte se logró mejorar la complejidad a $O(|E| + |V| \log |V|)$, usando en concreto un *heap* binario [14]. En el estudio que realizo utilizo binary heaps.

Por otro lado el algoritmo simplex es No Polinomial en el peor de los casos, ya que para todo $d > 1$ existe un problema lineal con $2d$ ecuaciones, $3d$ variables, y coeficientes enteros con valores absolutos acotados por 4, tales que simplex puede tomar $2^d - 1$ iteraciones para encontrar el óptimo [2].

Podemos decir entonces que la complejidad de simplex es superior a dijkstra, ya que incluso tomando dijkstra sin lista de prioridad, obtendremos $O(n^2) < O(2^d)$.

IV. Rendimiento

Para llevar acabo los experimentos utilizo una computadora GNU/Linux Debian 8 (x86-64), con un

[1]Los *fibonacci heaps* son un desarrollo de las colas binomiales.

kernel 3.8.0-35 corriendo sobre un procesador Intel Core i7-3517U con cada CPU a 1.90GHz y 3.8 GiB de memoria. Los dos algoritmos fueron escritos en Python3.4 usando la librería NumPy para simplex. Se corrieron ambos algoritmos 20 veces sobre el gráfico de las redes del metro y del metrobús, con los siguientes resultados:

Cuadro I
RENDIMIENTO EN SEGUNDOS

Simplex	Dijkstra
1.946	0.045
1.988	0.044
1.954	0.043
1.973	0.052
2.003	0.044
1.976	0.051
1.951	0.059
1.972	0.044
1.975	0.043
1.962	0.044
1.987	0.043
1.941	0.044
1.941	0.039
1.980	0.058
1.982	0.043
1.984	0.044
1.975	0.044
1.930	0.055
1.947	0.044
1.946	0.043

Usando una prueba de hipótesis t de dos muestras, donde nuestra hipótesis nula es que la media de las dos muestras es igual, se encontró que con un 95 % de confianza, que H_0 se rechaza, lo cual significa que la diferencia de medias es significativa. La media de Dijkstra fue de 0.0498 y la de simplex de 1.9656, con un intervalo de confianza de 1.904 a 1.927. Claramente la media de Dijkstra no entró en este intervalo y el tiempo de ejecución de simplex es mayor.

```
Two Sample t-test

data:  d$simplex and d$dij
t = 333.2444, df = 38,
p-value < 2.2e-16
alternative hypothesis:
true difference in means is
greater than 0
95 percent confidence interval:
 1.906157       Inf
 sample estimates:
mean of x mean of y
  1.96565   0.04980
```

V. CONCLUSIONES

Simplex es un algoritmo general para resolver problemas lineales, lo cual lo hace muy poderoso. Sin embargo, su complejidad es No Polinomial($O(2^d)$ en el peor caso), por lo que es preferible buscar otros algoritmos para casos específicos [2]. Si bien Simplex puede resolver el problema del camino mas corto, Dijkstra, un algoritmo especializado puede resolver esta tarea en un tiempo polinomial ($O(n^2)$ [1] en el peor caso sin optimizaciones). Esto también se demuestra en la práctica, donde se puede ver que el tiempo de ejecución y mediante una muestra de 20 ejecuciones se comprobó que simplex tarda más en ejecutarse que Dijkstra.

El resultado final de la ruta mas corta entre metro Cuatro Caminos y Tláhuac fue: Tláhuac, Tlaltenco, Zapotitlán, Nopalera, Olivos, Tezonco, Periférico Oriente, Calle 11, Lomas Estrella, San Andrés Tomatlán, Culhuacán, Atlalilco, Mexicaltzingo, Ermita, Portales, Nativitas, Villa de Cortés, Xola, Viaducto, Chabacano, San Antonio Abad, Pino Suárez, Isabel la Católica, Salto del Agua, Balderas, Juárez, Hidalgo, Revolución, San Cosme, Normal, Colegio Militar, Popotla, Cuitláhuac, Tacuba, Panteones, Cuatro caminos. Con una distancia de 31399 mestos.

REFERENCIAS

[1] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, no. 1, pp. 269–271, 1959.

[2] C. Papadimitriu and S. Kenneth, *Combinatorial Optimization. Algorithms and Complexity*, 2nd ed., Dover, Ed. United States: DOver, 1998.

[3] Z. R. Königsberg, "A generalized eigenmode algorithm for reducible regular matrices over the max-plus algebra with applications to the metro-bus public transport system in mexico city," *Nonlinear Analysis: Hybrid Systems*, vol. 2, no. 4, pp. 1205 – 1216, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1751570X0800054X

[4] M. Niksirat, M. Ghatee, and S. M. Hashemi, "Multimodal k-shortest viable path problem in tehran public transportation network and its solution applying ant colony and simulated annealing algorithms," *Applied Mathematical Modelling*, vol. 36, no. 11, pp. 5709 – 5726, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0307904X12000224

[5] C. de Desarrollo Tecnológico, "Longitud de estación a estación por linea." [Online], accesible: http://www.metro.df.gob.mx/operacion/longestaciones.html, Noviembre 2015.

[6] S. de Corredores de Transporte Público de Pasajeros del D.F., “Metrobús,” [Online], accesible: http://www.metrobus.df.gob.mx/, Noviembre 2015.
[7] R. Chandrasekaran, *Network Flows and Combinatorial Optimization*, 1st ed., U. Dallas, Ed. United States: UT Dallas, 1996.
[8] A. Schrijver, “On the history of the shortest path problem,” *Documenta Mathematica*, vol. 15, pp. 155–167, 2010.
[9] T. G. Heineman, G. Pollice, and S. Selkow, *Algorithms in a Nutshell*, 1st ed. O’Reilly Media Inc., Octubre 2008.
[10] R. Bellman, “On a routing problem,” *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.
[11] L. R. Ford Jr., *Network Flow Theory*. Santa Monica, California: RAND Corporation, Agosto 1959.
[12] J. Mager, “El algoritmo fringe search como solución superior a a* en la búsqueda de caminos sobre gráficos de malla.” Master’s thesis, Universidad Nacional Autónoma de México, Ciudad de México, México, 6 2015.
[13] M. L. Fredman and R. E. Tarjan, “Fibonacci heaps and their uses in improved network optimization algorithms,” *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987. [Online]. Available: http://doi.acm.org/10.1145/28869.28874
[14] M. Barbehenn, “A note on the complexity of dijkstra’s algorithm for graphs with weighted vertices,” *IEEE Transactions on Computers*, vol. 47, no. 2, p. 263, 1998.