Graph Theory in Global Positioning System (GPS)

Sharon Loh (13510086)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510086@std.stei.itb.ac.id

Abstract* — In this globalization era, we can't ignore that communication and transportation take a very important role in our daily routines. And now, technology has made a significant impact in improving transportation efficiency. GPS (Global Positioning System) navigation device is one of the latest technological trends which can be used to determine an object's position and find a direction and distance to other locations. Most people nowadays use this GPS device to determine the shortest path to the desired destination. This paper will discuss how graph theory can be used in determining the shortest path in GPS navigation devices.

***Index Terms* — GPS, Graph Theory, Shortest Path**

I. Introduction

The uncertainty in position has been the main problem for all human being in years. As time goes by, technology become more and more advance and it also provide modern devices which can be used to solve many problems. One of those devices is Global Positioning System navigation device, which is usually called as GPS. People no longer need a traditional map, which is sometimes confusing as we don't know exactly where our position is. Instead of that, people nowadays prefer to use GPS for helping them in determining the shortest path to their desired destination.

II. Graph Theory

A. The Definition of Graph

In computer science and mathematics, graph is one of the most common and important data structure. Graph refers to a set of vertices and edges. Vertices may represent a state or in this case, a position; while edge represents a relation between two vertices. Mathematically, graph is defined as G = (V , E) where

- V = a non-empty set whose elements are called vertices or nodes = $\{v_1, v_2, \dots v_n\}$
- E = a set of ordered pairs of vertices, called edge or arcs = $\{e_1, e_2, \dots , e_n\}$

Figure 2.1 below is an example of graph G where

- V = $\{v_1, v_2, v_3, v_4\}$
- E = $\{e_1, e_2, e_3, e_4, e_5\}$

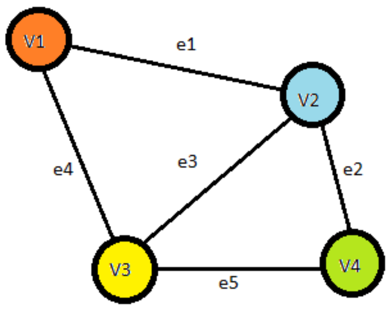

Figure 2.1 Example of a Graph

B. Directed Graph (digraph)

A graph whose edges are represented as arrows as it has a direction is called directed graph or digraph.

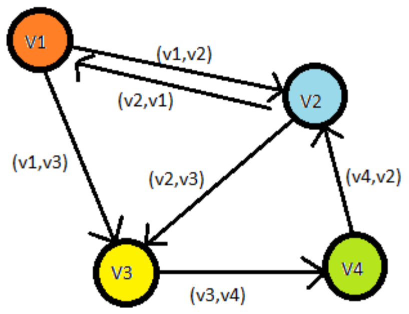

Figure 2.2 Directed Graph

In directed graph, (v_1, v_2) is not equal to (v_2, v_1). In edge (v_1, v_2), v_1 is called as the initial vertex while v_2 is the terminal vertex.

In this case, directed graph is used to represent the road network in GPS, where vertices represent the places and edges represent the roads. In the real word, the road can be either one way or two way traffic. That's why we use directed graph to represent the road path.

C. Weighted Graph

A weighted graph is a graph in which each edge is given a numerical weight. The weight of each edge represents the distance and the traffic condition (whether it is crowded or not) between two places (two vertex). Here is the example of a weighted graph.

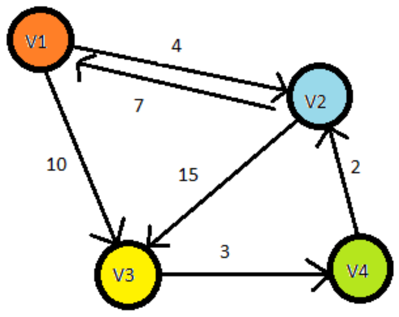

Figure 2.3 Weighted Graph

As you can see in the figure above, there are some numbers written on each edge. This numbers represent the weight (the distance and the traffic condition) of each edge (road). The bigger the weight, the more time is needed to move from one place to another through that road. So, the GPS tends to choose the path with a lower weight so that the system will find the shortest path.

III. GLOBAL POSITIONING SYSTEM

Global Positioning System (GPS) is a satellite-based positioning, timing, and navigation system made up of a network of 24 satellites placed into orbit by the U.S. Department of Defense. GPS is characterized as a satellite navigation that provides signals for location and time information. The precise time and stable frequency signals available from GPS are equal in importance to its navigation functions.

GPS is a stunning example of a technological innovation. It becomes the first and only global three-dimensional radio navigation and timing system. Originally, GPS was basically intended for military purposes. But then, since 1990s, GPS's services have been globally used for many other purposes. These include telecommunications, geo-tagging, map-making, robotics, electrical power distribution, banking and finance, transportation, emergency services, and military operations, and many more.

For navigation purposes, GPS is used to determine the user's position and how far the distance is from the nearest destinations. It is now a great device used by anyone who has the need to navigate either great or small distances.

A. *Basic Principles*

The basis of the GPS technology is a set of 24 satellites that orbit the earth once every 12 hours in a very precise orbit. The satellites which are powered by solar energy transmit radio signal information to earth as to the exact time and location. GPS receiver takes this information. Once the GPS receiver locks on to at least three satellites, it can use the triangulation concept to calculate the 2D user's exact position (latitude and longitude) and track movement. And if it locks on four or more satellites, it can determine the user's 3D position (latitude, longitude, and altitude). The GPS receiver compares the time when a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. And with the distance measurements from at least four satellites, the receiver can determine the user's position and display it on the unit's electronic map. Then, once the user's position has been determined, the GPS unit can calculate other information, such as speed, track, distance to destination, and more.

B. *GPS Structure*

GPS consists of three segments :

- space segment
- control segment
- user segment

The space segment consists of 24 satellites in 6 orbital planes with 4 satellites each. It takes each satellite about twelve hours to orbit the earth. Each satellite is in its own orbit 11.000 miles above the Earth.

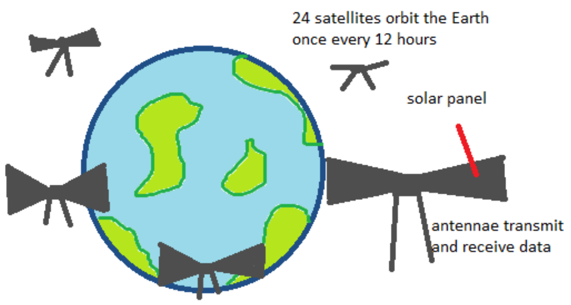
24 satellites orbit the Earth once every 12 hours
solar panel
antennae transmit and receive data

Figure 3.1 Space Segment

The control segment is a group of ground stations that monitor and operate the GPS satellites. The control segment consists of a Master Control Station (MSC), an alternate master control station, four ground antennas, and six monitor stations. Each station will send the information to the Master Control Station which then updates and corrects the navigational message of the satellites.

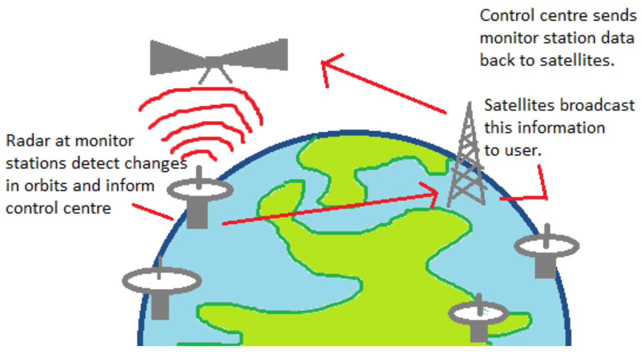
Control centre sends monitor station data back to satellites.
Satellites broadcast this information to user.
Radar at monitor stations detect changes in orbits and inform control centre

Figure 3.2 Control Segment

The user requires a GPS receiver in order to receive the transmissions from the satellites. The GPS receiver calculates the location based on signals from the satellites. User does not transmit anything to the satellites and therefore the satellites don't know the user is there. The only data the satellites receive is from the Master Control Station (which is located in Colorado). Users consist of both the military and civilians.

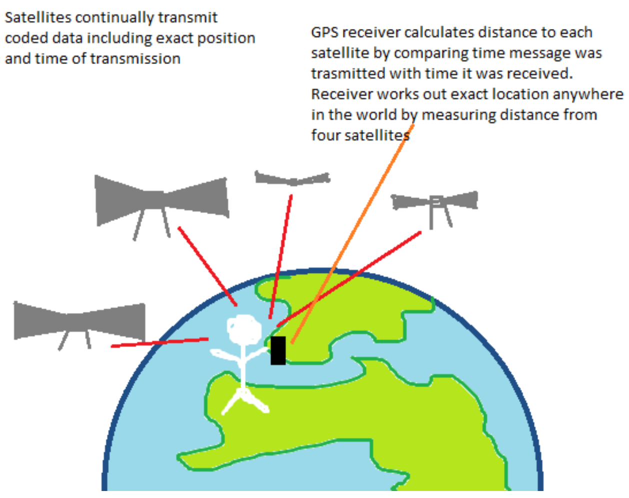
Satellites continually transmit coded data including exact position and time of transmission
GPS receiver calculates distance to each satellite by comparing time message was trasmitted with time it was received. Receiver works out exact location anywhere in the world by measuring distance from four satellites

Figure 3.3 User Segment

IV. THE SHORTEST PATH ALGORITHM

Since what we actually discuss in this paper is the use of GPS for navigation system, we need to know more about how GPS determine the shortest path between two places.

There are 4 well-known algorithm which is used to determine the shortest path between vertices. They are : Dijkstra's Algorithm which is used to solve the single-source shortest path problem, Floyd-Warshall Algorithm which is used to solve the all-pairs shortest path problem, Bellman-Ford algorithm which is used to find the one-source shortest paths, and Johnson's Algorithm which is commonly used to find the shortest paths between all pairs of vertices in a sparse digraph.

A. *Dijkstra's Algorithm*

Djikstra's algorithm (named after its discover, E.W. Dijkstra) is a greedy algorithm that solves the problem of finding the shortest path from a point in a graph (the source) to a destination. It functions by constructing a shortest-path tree from the initial vertex to every other vertex in the graph. Even so, it only works on graphs which have no edges of negative weight. In this case, vertices of the graph represent places and edge is the distance between two paired places.

The actual speed of the algorithm can vary from O(*n*log(log(n))*) to O(*n^2*).

First, let us define the node at which we are starting (the initial node) as S. And the graph whose weight of every vertices we calculate is G.

The algorithm is as follow, starting from node S :

1. Set the distance to all vertex as infinite
   Mark all vertex as unvisited

```
for each vertex V in Graph do
   d[v] = INFITINTY
   visit[v] = false
```

2. Set the distance to source S to 0

```
d[S] = 0
```

3. Iterate over all vertices from 1 to v to find the vertex u which has the smallest distance and has not been visited yet

```
//u = -1 means that vertex u hasn't been
choose before
u = -1
for i=1 to v do
  if (u=-1 or d[u]>d[i]) and (visit[i]=false)
     u = i
  end if
end loop
```

4. After choosing the vertex u, mark u as visited.
   Then, calculate the distance for all other neighbors of vertex u.

```
visit[u] = true
for i=1 to v do
    if (weight[u][i] > -1)
       d[i]= min(d[i],d[u]+weight[u][i])
    end if
 end loop
```

5. Repeat step 3 and step 4 until all vertices are visited.

Dijkstra's Algorithm is commonly used to solve the single-source shortest path problem, in which we have to find shortest paths from a source vertex v to all other vertices in the graph.
Note that the edge weight may not be a negative value.

B. *Floyd – Warshall Algorithm*

The Floyd-Warshall algorithm is also variously known as Floyd's algorithm, the Roy-Floyd algorithm, the Roy-Warshall algorithm, or the WFI algorithm. It is an algorithm for efficiently finding the shortest paths between every pair of vertices in a weighted directed graph.

The Floyd–Warshall algorithm compares all possible paths through the graph between each pair of vertices. That's why the worst case algorithm complexity is O(n3).

The Floyd – Warshall algorithm is as follow.

1. Let `d` be a `N×N` matrix where `N` is the number of vertices. Here, `d[i,j]` represents the length of the shortest path from `i` to `j`.
   For each element `d[i,j]`, assign a value equal to the weight between vertex `i` and vertex `j`, or assign is as infinity value if the edge between `i` and `j` does not exist.

```
for i = 1 to N
   for j = 1 to N
      if there is an edge from i to j
         d[i][j] = weight[i,j]
       else
         dist[i][j] = INFINITY
```

2. For each pair of vertices `i` and `j`, see if there's a vertex `k` so that the path from `i` to `j` through `k` is shorter than the one already found for `i` and `j` which is `d[i,j]`.

```
for k = 1 to N
   for i = 1 to N
      for j = 1 to N
         if (d[i,k]+d[k,j] < d[i,j]) then
            d(i,j) = d(i,k) + d(k,j)
```

This will give the shortest distances between any two nodes, from which shortest paths may be constructed.
But as we can see, there are 3 nested loop each being executed N times, thus the complexity of the algorithm is O(n^3).

This algorithm can also calculate the distance between two vertices in which the weight of the edge between those 2 vertices has a negative value. But note that in this case, 2 vertices are 2 places and the weight between those 2 vertices is the distance and the traffic condition between place 1 to place 2, so in this case, distance may not be a negative value.

C. *Bellman – Ford Algorithm*

Bellman – Ford algorithm was developed by Richard Bellman and Lester For, Jr. This algorithm is used to compute a single-source shortest path in a weighted directed graph. Bellman–Ford is actually used primarily for graphs with negative edge weights.

The algorithm is as follow.

1. Set the shortest path to all nodes, unless the source, as infinity. Mark the length of the path to the source as 0.

```
for each vertex v in Graph do
    d[i] <- INFINITY;
d[s] <= 0
```

2. Relax each edge of the graph. Relaxing an edge means checking to see if the path to the node the edge is pointing to can't be shortened, and if so, do it. Note that we can't relax the edges whose start has the shortest path of length infinity to it. Apply this step n-1 times where n is the number of vertex v in the graph.

```
for i=0 to n-1 do
  for j=0 to e do
     if (d[ed[j].u]+ed[j].w < d[ed[j].v])
        d[ed[j].v] = d[ed[j].u] + ed[j].w
```

Here, d[i] is the shortest path to node I, while e is the number of edges and ed[i] is the ith edge.

The basic structure of Bellman–Ford algorithm is very similar to Dijkstra's algorithm, but instead of greedily selecting the minimum-weight node, it simply relaxes all the edges, and does this n − 1 times. The complexity of this algorithm is O(|v||e|) where |v| is the number of vertices of the graph and |e| is the number of edges.

D. *Johnson's Algorithm*

In 1977, Donald B. Johnson has defined an algorithm that takes advantage of the sparse nature of most graphs, with directed edges and no cycles. It is used to find the shortest path from every vertex to every other vertex, or we can simply call it as all-pairs shortest paths. Johnson's algorithm works based on Bellman-Ford algorithm and Dijkstra's algorithm.

The Johnson's algorithm is as follow.

1. Add a new node (for example S) with 0 weight edge from it to all other nodes
2. For each node, run the Bellman – Ford algorithm once starting from the new node we created which was named as S so that we can find each vertex v with the least weight from S to v. Algorithm is terminated if a negative cycle is detected. But in this case, there are no negative weight or even negative cycle, since the weight of two vertices represent the distance and the traffic condition between two places. The weight from vertex S to v is named as d[v].

3. Reweight every edge of the original graph using the values computed by the Bellman-Ford algorithm. An edge from u to v which have the weight w(u,v) is given the new weight value w(u,v) + d(u) − d(v).
4. Remove S. Then, run Dijkstra's algorithm to find the shortest paths from each vertex to every other vertex in the reweighted graph.

When the graph is sparse, the total time can be faster than the Floyd–Warshall algorithm, which solves the same problem in time $O(V^3)$.

V. CONCLUSION

Global Positioning System or commonly known as GPS is one of the latest technological developed device which use graph theory (the algorithm of finding the shortest paths in graph) to be able to find a path between one place to another. The algorithm which is well-known in solving the shortest path problems are Dijkstra's Algorithm, Floyd-Warshall Algorithm, Bellman-Ford algorithm, and Johnson's Algorithm. Each of which has their own characteristic.

REFERENCES

[1] Munir,Rinaldi, Diktat Kuliah IF2091 Struktur Diskrit, 4th Edition, Institut Teknologi Bandung, Bandung

[2] http://en.wikipedia.org/wiki/Global_Positioning_System access 8th December 2011

[3] http://searchmobilecomputing.techtarget.com/definition/Global-Positioning-System access 8th December 2011

[4] http://en.wikipedia.org/wiki/Graph_(mathematics) access 8th December 2011

[5] http://en.wikipedia.org/wiki/Graph_theory access 8th December 2011

[6] http://en.wikipedia.org/wiki/Global_Positioning_System access 9th December 2011

[7] http://en.wikipedia.org/wiki/Dijkstra's_algorithm access 9th December 2011

[8] http://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm access 9th December 2011

[9] http://en.wikipedia.org/wiki/Johnson's_algorithm access 9th December 2011

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11th December 2011

[illegible]

Sharon Loh (13510086)