Contents

1 Reference Material

This section records information for easy reference.

2 Problem Description

In the real world, we are facing a lot of optimization problem. For example: A saleman would like to travel all the cities in the Great Toronto Area to promote companies products. This person does not want to visit the city two times. What is the shortest possible route that visits each city exactly once? This problem can also apply to logistics and telecommunication. Find the most optimized solution usually will bring the cost down.

3 General System Description

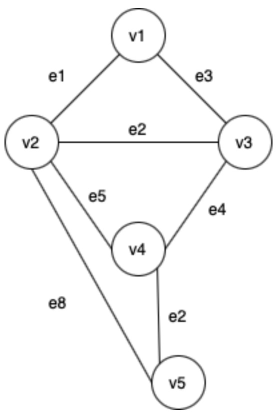
4 Specific System Description

4.1 General Definition

- v vertex: a basic unit(such a city) disperse in a graph.
- V vertices: a set of vertices.
- e edge: a connction bewteen two vertices.
- E edges: a set of edges.
- G graph: A graph G is an ordered triple (V, E, f) consisting of a nonempty set V of vertices, a set E of edges, and an incidence function f that associates with each edge and two vertices.
- f relation function bewteen e and v: If e is an edge and u and v are vertices, u, v $\in$ V, such that f(e) = u$\sim$v, then e is said to join u and v; u and v are endpoints of edge e.
- G = (V, E, f, w) is a weighted graph: A weighted graph is a graph with each edge e of G let there be associated a real number w(e).

- w: E→ $\Re$ Weight: With each edge e of G let there be associated a real number w(e).
- ve: either a vertex or an edge
- VE: a set of ve

4.2 Graph in Drawing



4.3 Example of a Weighted Graph

- G = (V(G), E(G), f, w)
- V = $\{v_1, v_2, v_3, v_4, v_5\}$
- E = $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$
- f(e_1) = $v_1 \sim v_2$, f(e_2) = $v_2 \sim v_3$, f(e3) = $v_1 \sim v_3$, f(e4) = $v_3 \sim v_4$,
- f(e5) = $v_2 \sim v_4$, f(e_6) = $v_4 \sim v_5$, f(e_7) = $v_2 \sim v_5$, f(e_8) = $v_2 \sim v_5$
- w(e_1) = n_1, w(e_2) = n_2, w(e_3) = n_3, w(e_4) = n_4,
- w(e_5) = n_5, w(e_6) = n_6, w(e_7) = n_7, w(e_8) = n_8 where $n \in \Re$

4.4 Goal Statements

Give a weighted graph, the goal statements are:

- Find the shortest path where all vertices appear only once in a path.
- Find the total weight of the shortest path.

4.5 Assumptions

- V and E are finite sets.
- It is undirected graphs, means f(e) = u$\sim$v = v$\sim$u.
- It is a connected graph.
- There is no loop in the graph.

4.6 Theoretical Model

Minimize({p:Path | p$\in$SP: TotalWeight(p)})

- SetVE $\equiv V||E$
- AllCombinaiton $\equiv \cup(x : ve|x \in SetVE : \{Combine(x)\})$
- Combine(a,b) $\equiv$ [a||b]
- SW :=

  $\{x, i : VE, N|x \in AllCombinaiton \wedge 1 \leq i \leq length(x) \wedge f(x[i]) = x[i-1] \sim x[i] : x\}$
- length(x) $\equiv$ +(y:VE |y$\in$x: 1)
- SP :=

  $\{w : Walk|w \in SW : \forall(x : Walk|x \in SW : IsDistinctV(x) \wedge IsDistinctE(x) \wedge IsVisitAllV(x))\}$
- IsDistinctV(x) $\equiv$ (+(v:vertice | v$\in$x: 1) == +(v:vertice | v$\in$SetofVertice(x): 1))
- IsDistinctE(x) $\equiv$ (+(e:edge | e$\in$x: 1) == +(e:edge | e$\in$SetofEdge(x): 1))
- IsVisitAllV(x) $\equiv$ (+(v:vertice | v$\in$V: 1) == +(v:vertice | v$\in$SetofVertice(x): 1))
- SetofEdge(x) $\equiv \cup$(e:edge | e$\in$E: {e})
- SetofVertice(x)$\equiv \cup$(v:vertice | v$\in$x: {v})
- TotalWeight(p) $\equiv$ +(e:edge | e$\in$p: w(e))
- Minimize(TWs) $\equiv (w \in TWs) \wedge \forall(x : R|x \in TWs : w \leq x)$

4.7 Supporting Data Definitions

- A graph G is connected if for every u, v $\in$ V there exists a path connect u and v in G. Otherwise G is called disconnected.
- Finite Graph: A graph is finite if both its set of vertices and set of edges are finite.
- Endpoint: if f(e) = u$\sim$v, edge e has endpints of u and v. u and v $\in$ V(G).
- Loop: An edge with identical ends. f(e) = vu and v = u. u and v $\in$ V(G).
- Connected Graph: if for every u, v $\in$ V there exists f(e) = u$\sim$v. Otherwise G is called disconnected.
- Walk: A walk is a well formed sequence W = $< v_0, e_1, v_1, ..., e_k, v_k >$, whose terms are alternatively vertices and edges such that for 1$\leq$i$\leq$k, the edge e_i has endpoints v_{i-1} and v_i. f(e_i) = $v_{i-1}v_i$. k is the length of W.
- Trail: If the edges e_1, e_2, ..., e_k of a walk are distinct.
- Path: a path is a walk if both of its edges e_1, e_2, ..., e_k and its vertices v_0, v_1, ... , v_k are distinct.

4.8 Instance Models

- Input: G = (V, E, f, w)
- Output: a sequence

5 Likely Changes

- There could be new algorithms to solve graph problems

6 Unlikely Changes

- The final output is the most optimal result, it is unlikely change.

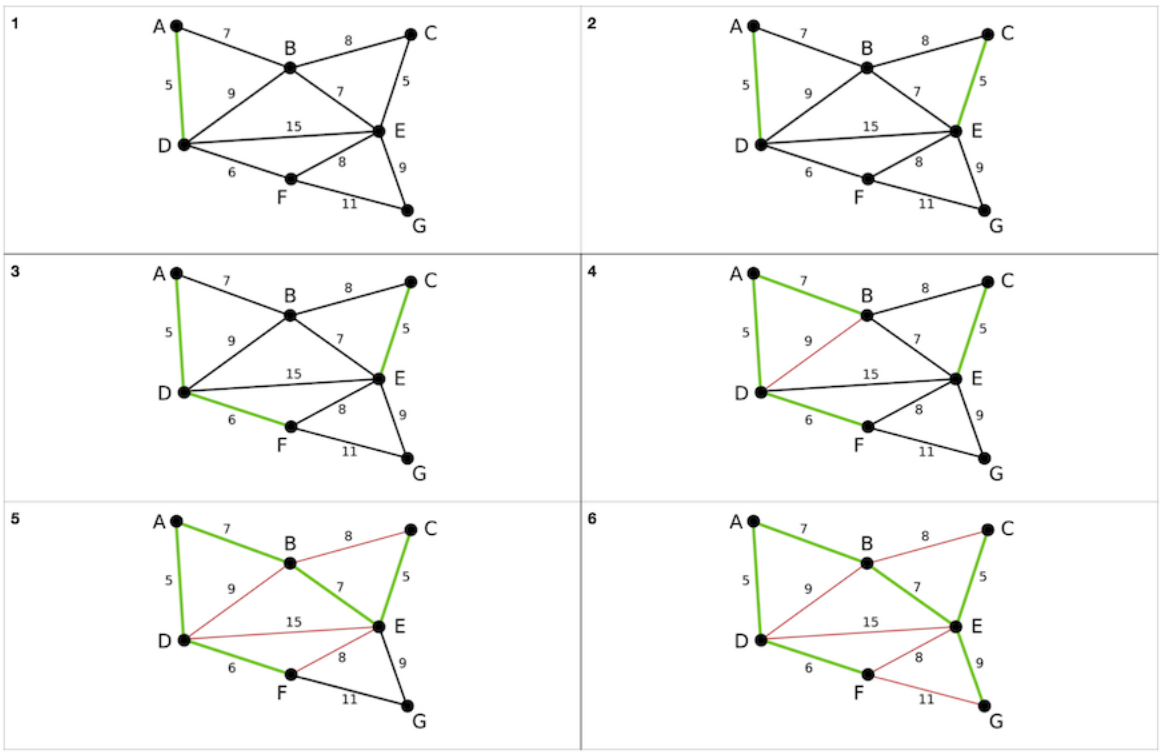
7 Algorithm

7.1 Kruskal's Algorithm

Given G = (V, E, f, w)

- initialize F = (V, $\emptyset$)
- for edges e in increasing order of length w
- if edges e are do not form a cycle with F
- $\ldots\ldots$ {add e to F}
- return F

7.2 Pseudo Code in Drawing



7.3 Nearest Neighbor Alorithm(usually can't find an optimized solution)

8 Question

- well formed sequence such as

9 Reference

- Graph Theory With Applications by J.A. Bondy and U. S. R. Murty

- Nearest Neighbor Alorithm
- Kruskal Alorithm in TSP