Algorithm on Tape Graph and Their Geometric Transformations

M.EL-Ghoul
Mathematics Department, Faculty of Science
Tanta University, Tanta
Egypt

M. M. Khalil
Mathematics Department, Faculty of Science
Al Azhar University
Egypt

H. Ahmed
Mathematics Department, Faculty of Girls
Ain Shams University
Egypt

Abstract

In this paper, we introduce the algorithm on tape graph when the tape graph weighted and unweighted, and we discusses some geometric transformations (folding, deformation and retraction) on the tape graph, when the tape graph weighted and unweighted.

Keywords: Algorithm, tape graph, deformation.

Definitions and Background:

(1)Graph:

A graph is intuitively a finite set of points in space, called the vertices of the graph, some pairs of vertices being joined by arcs, called the edges of the graph [3].

(2) Tape graph:

The tape graph G is a diagram consisting of a finite non empty set of the elements with "line or curve" shape called "vertices" dented by V (G) together with elements, with "tape" shape called "edges" denoted by E (G) [8].

(3) Algorithm:

In mathematics and computer science, an algorithm is an effective method expressed as a finite list o 2010 Mathematics subject classification 5C5, 68 R 10.Well-defined instructions for calculating a function. In simple words an algorithm is a step-by-step procedure for calculations [5].

(4) Graph Algorithm:

Graph algorithms are one of the oldest classes of algorithms and they have been studied for almost 300 years (in 1736) which solve problems related to graph theory. There are some of important algorithms for solving these problems [5].

(5) Kruskal's algorithm:

In Kruskal's algorithm, the edges of weighted graph are examined one by one in order of increasing weight. At each stage the edge being examined is added to what will become the minimum spanning tree, provided that this addition doesn't create a circuit. After n -1 edges have been added (where n is the number of vertices of the graph), these edges, together with the vertices of the graph form a Minimum spanning tree for the graph [4].

(6) Language of the algorithm:

How it works:

Input: G (a weighted graph with n vertices).

Algorithm body:

(Build a subgraph T of G to consist of all the vertices of G with edges added in order of increasing weight. At each stage, let m is the number of edges of T).

1. Initialized T to have all vertices of G and no edges.

2. Let E be the set of all edges of G, and let m: =0.

[Pre-condition: G is connected].

3. While (m≤n-1).

3a. Find an edge e in E of least weight.

3b.Delete e from E.

3c. If addition of e to edge set of T doesn't produce a circuit.

Then add e to the edge set of T and set m: =m+1 [4].

(7) Folding

The field of folding began with S.A.Robertson's work , in 1977 , on isometric folding of Riemannian manifold M into N , which send any piecewise geodesic path in M to a piecewise geodesic path with the same length in N [2].

(8) Deformation

The deformation of geometric tape graph is the change in the metric properties of a continuous tape graph G in the displacement from an initial placement G_0 to a final placement G. A change in the metric properties means that the edges and vertices drawn in the initial tape graph placement changes its length of edges and vertices when displaced to a tape graph in the final placement. So the deformation is a change in the shape or size of a tape graph due to an applied force [8].

(9) Retracts

A subset A of a topological space X is called a " retract" of X if there exists a continuous map $r:X \rightarrow A$ (called a retraction) such that r(a)=a for all $a \in A$, where A is closed and X is open [9].

(10) Null graph:

The null graph is a graph consists of a set of vertices and no edges [10].

The main results:

Aiming to our study, we will introduce the algorithm on the tape graph, we applied algorithm on the tape graph with geometric transformations. (Folding, deformation and retraction.

Algorithm in tape graph:

We will compute the algorithm on the tape graph by two types.

Type (1) unweighted tape graph:

Let G be a tape graph with two vertices v_0,v_1 and one edge e_0 and dim =2, we can compute its algorithm as follows:

Input: G (tape graph with vertices v_0,v_1). Algorithm body: Build a subgraph T of G to consist of all the vertices of G with edges added in order of increasing weight, e_0 edge of T.

1. Visit v_0.

2. Attach the edge e_0 to T.

3. Visit v_1.
End while.

4. Output T.
End algorithm.

Shown in Fig.(1.1).

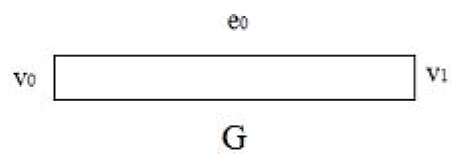

Fig.(1.1)

Type (2) weighted tape graph:

When we measure the distance of the tape graph $G((v_0 v_1)=\alpha=20)$.

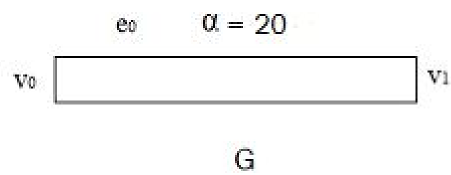

Fig.(1.2)

By using Kruscal's algorithm we find minimum spanning tree as follows:

Iteration no	Distance considered	Weight	Action taken
1	v_0-v_1	20	Added

Minimum spanning tree is the same figure (1.2).

Folding the tape graph:

Type (1) unweighted folding tape graph:

The first case :(edge to itself) Let $f_1 : G_1 \to G_2$, $f_i : G_2 \to G_3$.

We can compute its algorithm as follows:
Input:

G_1(tape graph with vertices v_0, v_1).

Algorithm body:
Build a graph G_2, G_3 to G_1 by geometric transformations (folding $f_1, f_{ii} \to \infty$).

1. Visit v_0.
2. Attach the edge e_0 go to itself.
3. Go to step 1 for the other vertex v_1 and the edge e_0.
End while.
4. Output G_3 (simple graph).
End algorithm.

Shown in Fig.(1.3).

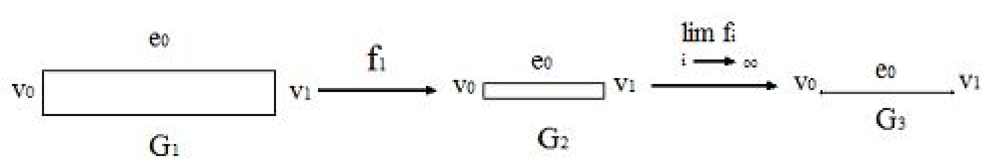

Fig.(1.3)

The second case :(vertex to vertex)

Let $f: G_1 \to G_2$. We can compute its algorithm as follows:

Input:

G_1 (tape graph with vertices v_0, v_1).

Algorithm body: Build a graph G_2 to G_1 by geometric transformations (folding f_1).

1. Visit v_0.
2. Attach the edge e_0 to G_2.
3. Go to step 1 for the other vertex v_1 and this vertex go to v_0.

End while.

4. Output G_2.

End algorithm.

Shown in Fig.(1.4).

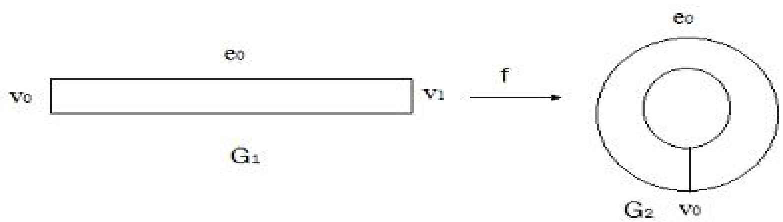

Fig.(1.4)

Type (2) weighted folding tape graph:

The first case :(edge to itself)

Let $f_1 : G_1 \rightarrow G_2$, $f_i : G_2 \rightarrow G_3$.

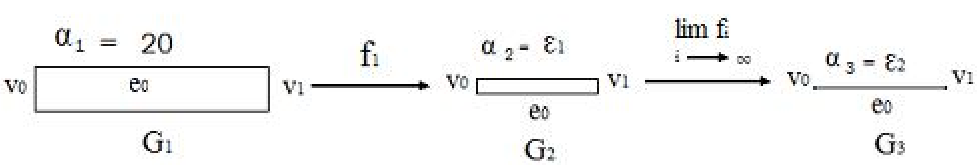

Fig.(1.5)

Where $\varepsilon_1 < 20, \varepsilon_2 < \varepsilon_1$.

By using Kruscal's algorithm we find minimum spanning tree as follows:

Iteration no	Edge considered	Weight	Action taken
1	v_0-v_1	ε_2	Added

Minimum spanning tree is simple graph.

Shown in Fig.(1.6).

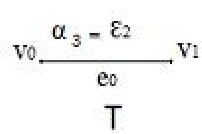

Fig.(1.6)

Theorem (1):

In the case edge to itself the folding in algorithm of weighted or unweighted tape graph go to weighted or unweighted simple graph.

Proof: The proof is clear from the above discussion.

The second case :(vertex to vertex).

Let $f:G_1 \rightarrow G_2$.

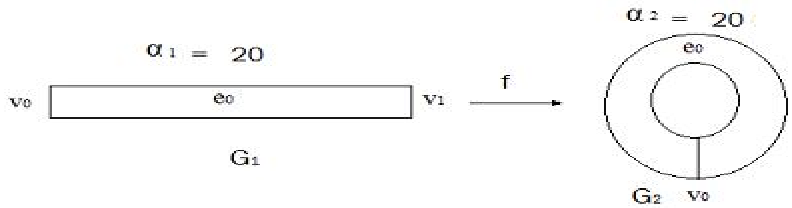

Fig.(1.7)

By using Kruscal's algorithm we find minimum spanning tree as follows:

Iteration no	Distance considered	Weight	Action taken
1	v_0-v_0	20	Not added

In this case there is no minimum spanning tree.

Theorem (2):

In the case weighted folding tape graph in case vertex to vertex there are no minimum spanning tree.

Proof: The proof is clear from the above discussion.

Deformation the tape graph:

Type (1) unweighted deformation tape graph:

Let $d_1:G_1 \rightarrow G_2$, $d_i:G_2 \rightarrow G_3$

We can compute its algorithm as follows:

Input:

G_1 (tape graph with vertices v_0,v_1).

Algorithm body: Build a graph G_2,G_3 to G_1 by geometric transformations (deformation $d_1,d_{ii} \rightarrow \infty$).

1. Visit v_0.

2. Attach the edge e_0 to G_2.

3. Go to step 1 for the other vertex v_1 and the edge e_0.

End while.

4. Output G_3 by d_i.

End algorithm.

Shown in Fig.(1.8).

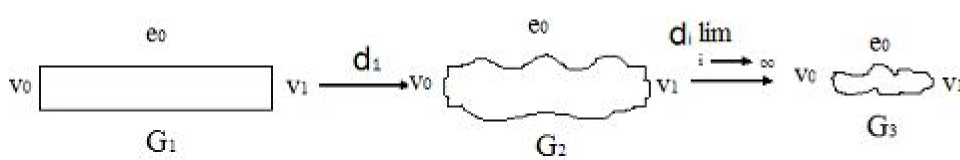

Fig.(1.8)

Type (2) weighted deformation tape graph:

Let $d_1:G_1 \rightarrow G_2$, $d_i:G_2 \rightarrow G_3$.

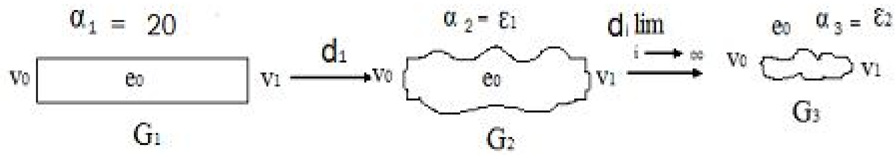

Fig.(1.9)

Where $\varepsilon_1 < 20, \varepsilon_2 < \varepsilon_1$.

By using Kruscal's algorithm we find minimum spanning tree as follows:

Iteration no	Distance considered	Weight	Action taken
1	α_3	ε_2	Added

Minimum spanning tree is:

Shown in Fig.(1.10).

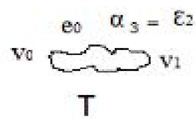

Fig.(1.10)

Retraction the tape graph:

Type (1) unweighted retraction tape graph:

The first case :(edge to itself)

Let $r:(G_1 - e_0) \rightarrow G_2$.

We can compute its algorithm as follows:

Input:

G_1 (tape graph with vertices v_0, v_1).

Algorithm body: Build a graph G_2 to G_1 by geometric transformations (retraction r).

1. Visit v_0.
2. Delete the edge e_0 from G_1.
3. Go to step 1 for the other vertex v_1 go to G_2.

End while.

4. Output G_2.

End algorithm.

Shown in Fig.(1.11).

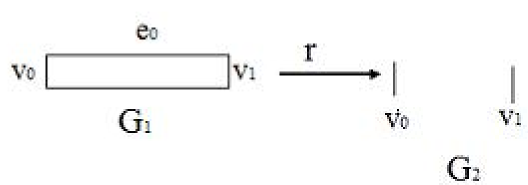

Fig.(1.11)

The second case :(vertex to vertex).

Let $r:(G_1 - v_1) \rightarrow G_2$.

We can compute its algorithm as follows:

Input:

G_1 (tape graph with vertices v_0,v_1).

Algorithm body: Build a graph G_2 to G_1 by geometric transformations (retraction r).

1. Visit v_0.
2. Delete the edge e_0 and the vertex v_1 from G_1.

End while.

4. Output G_2.

End algorithm.

Shown in Fig.(1.12).

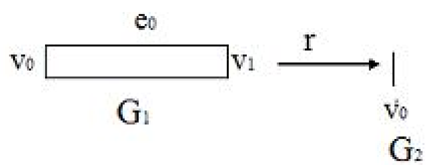

Fig.(1.12)

Type (2) weighted retraction tape graph:

The first case :(edge to itself)

Let $r:(G_1-e_0)\rightarrow G_2$.

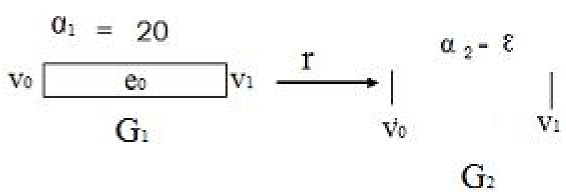

Fig.(1.13)

Where $\varepsilon<20$.

By using Kruscal's algorithm we find minimum spanning tree as follows:

Iteration no	Considered	Weight	Action taken
1	α_2	ε	Added

Minimum spanning tree is null graph.

Shown in Fig.(1.14).

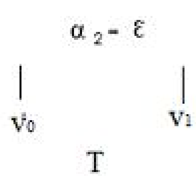

Fig. (1.14)

The second case :(vertex to vertex).

Let $r:(G_1-v_1)\rightarrow G_2$.

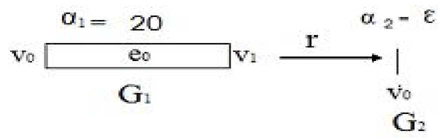

Fig.(1.15)

Where ε<20.

By using Kruscal's algorithm we find minimum spanning tree as follows:

Iteration no	Considered	Weight	Action taken
1	v_0-v_0	E	Added

Minimum spanning tree is null graph.
Shown in Fig.(1.16).

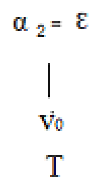

Fig.(1.16)

Theorem (3):

The retraction in algorithm of weighted or unweighted tape graph goes to weighted or unweighted null graph.

Proof: The proof is clear from the above discussion.

Applications:

1. Taenia (tapeworms) represent weighted tape graph.

2. The ruler is weighted tape graph.
3. Convert bar train represent unweighted tape graph and unweighted folding tape graph.

References

Balakrishnan V.K, "Schaum's Outline of theory and problems of graph theory". University of Maine. Orono, Maine, 1995.
El-Ghoul M.: Folding of fuzzy graphs and fuzzy spheres, Fuzzy Sets and Systems, Germany, 58.355-363, 1993.
Fournier, Jean-Claude, "Graph Theory and applications with Exercises and Problems", ISTE Ltd, 2009.
Susanna S.Epp, Discrete Mathematics with Application, Third Edition, Thomson Learning, Inc, 2004.
http://www.softpanorama.org/Algorithms/graph_algorithms.shtml#Hi story.
Giblin P.J.: Graphs, surfaces and homology, an Introduction to algebraic topology. Chapman and Hall Ltd, London. 1977.
Massey W.S.: Algebraic topology, an introduction. Harcourt, Brace& World, Inc, NewYork U.S.A. 1967.
M.El-Ghoul, M.M.Khalil. : Chaotic Tape Graphs, Journal of Mathematics Research in Canada.2011.
M.El-Ghoul, SH.Adel.Retraction of the new Graph.Journal of Applied Sciences Research in Pakistan.2009.
Wilson R.J.; Watkins J.J: Graph, an introductory approach, a fist course in discrete mathematics. Jon Wiley and Sons Inc, Canda.1990.