Scientific Journal of Impact Factor (SJIF): 4.14
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 6, June -2016

An Exon-Shuffling Genetic Algorithm for Degree-Constrained Minimum Spanning Tree

Sagar Jani[1],D.A.Parikh[2]

[1]*M.E (C.E), L.D.C.E, Ahmedabad*
[2]*Head, Computer Engineering Department, L.D.C.E, Ahmedabad*

Abstract:*Degree-Constrained Minimum Spanning Tree is a NP-Hard problem which has been tried to solve using different techniques to reduce its hardness. In this paper we have introduced a novel method to solve the problem of degree constrained minimum spanning tree.*

Keywords: Degree-Constrained Minimum Spanning Tree, Exon-Shuffling Genetic Algorithm.

I. INTRODUCTION

In graph theory, a spanning tree is a tree which has the total sum of edges to a minimum. Adding a degree constraint *k*, we can introduce a degree-constrained minimum spanning tree where each vertex has degree not more than *k*.

1.1 Formal definition [1]:

Input: *n*-node undirected graph G(V,E); positive integer $k \leq n$.

Question: Does G have a spanning tree in which no node has degree greater than *k*?

1.2 Degree Constrained Minimum Spanning Tree:

On a weighted graph, a Degree-constrained minimum spanning tree (DCMST) is a degree-constrained spanning tree in which the sum of its edges has the minimum possible sum. Finding a DCMST is an NP-Hard problem[2].

1.3 Exon Shuffling [3]:

The phenomenon of interrupted genes has been widely analysed and discussed ever since the discovery of the first introns. Roy (2003) provides an excellent review of the *ExonTheory of Genes*, the most important points of which are summarised here. Blake (1978) proposed that interrupted genes are essentially patched together from exons that code for simple protein structures, and Gilbert (1978) observed that introns could serve as buffers that allow the recombination of exons to create new protein products. Blake also proposed that this mechanism would be most efficient if the exons corresponded to independent units that determine discrete characteristics of a protein. Similarly, the exon shuffling hypothesis proposed by Gilbert views the evolution of genes as the recombination of independent units (exons) that code for independent protein structures. Numerous genes have been found where exons do indeed correspond to independent protein domains, and exon shuffling is likely to have played a vital role in the emergence of complex genes and other existing phenomena such as multi-cellularity.

II. PROPOSED METHODOLOGY

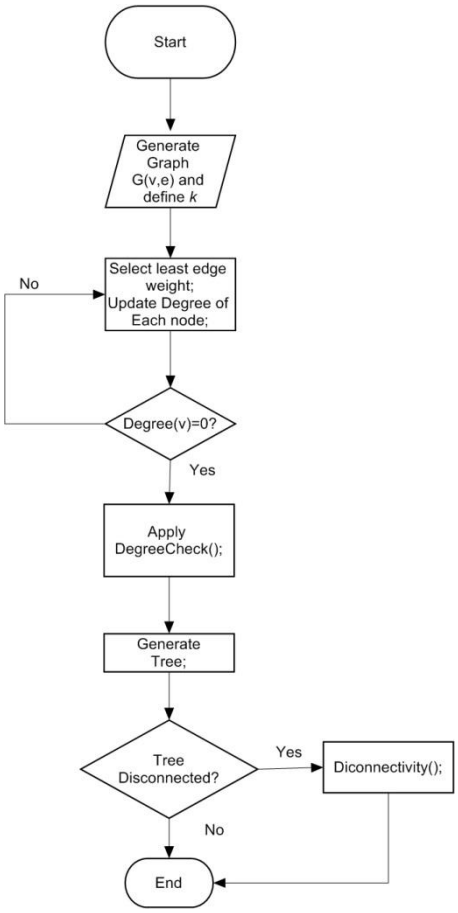
2.1 Overview of proposed methodology:

We are going to use the approach used in developing the exon shuffling genetic algorithm, we will perform the following steps:

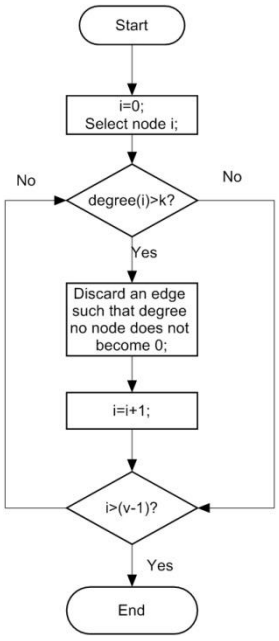
Step:1	Select a graph G with *v* vertices and *e* edges.
Step:2	Select the least edge weight.
Step:3	Select next least weight edge until all nodes are covered.
Step:4	If all nodes are covered check whether the tree obtained is connected or not.
Step:5	Generate the final tree.

2.2 Algorithm:

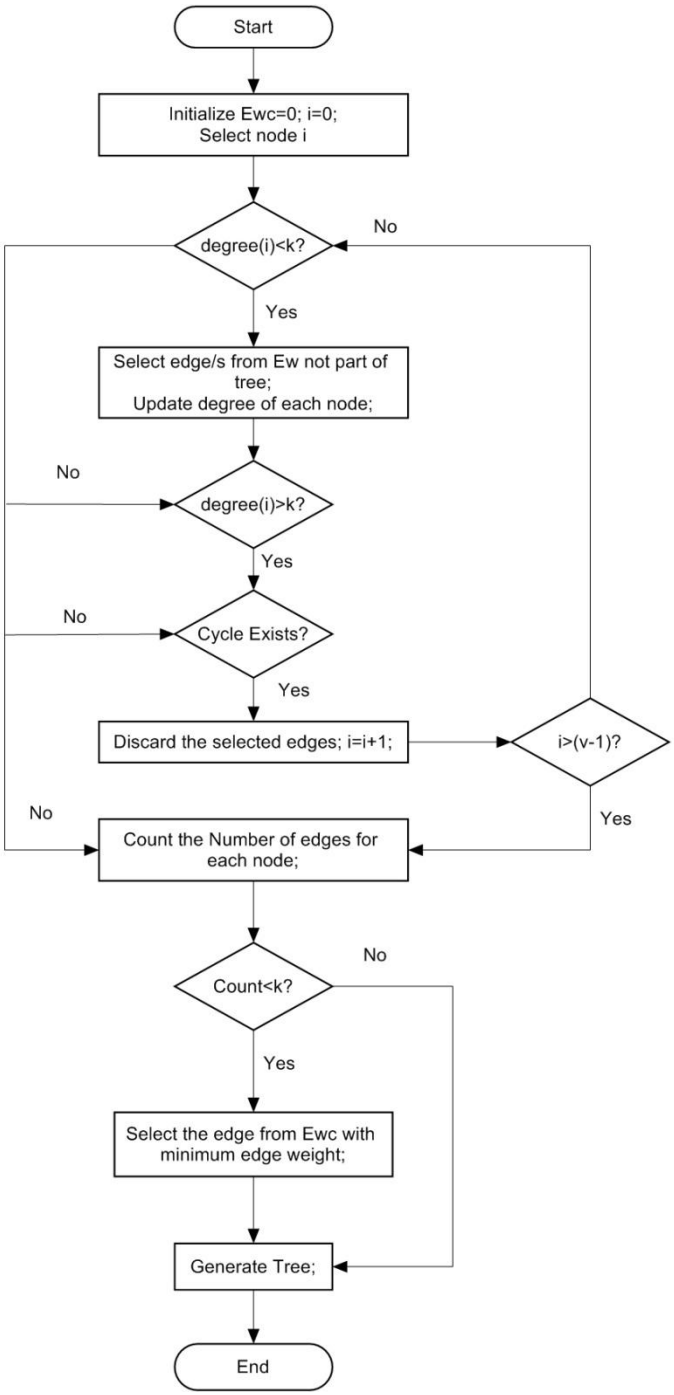
2.2.1Main Function:



2.2.2 DegreeCheck Function:



2.2.4Disconnectivity Function:



III. IMPLEMENTATION AND RESULT ANALYSIS

3.1 Implementation Parameters:

The implementation of the above algorithm requires considering a few parameters such as:

- All values should be positive.
- While we are considering the possibility of two or more than two or all edge weights to be same, to non-realize this consideration, we will be using time based random number generation for edge weights.
- The value of edge for the same node, we will take it as zero.
- Theoretically, there is a possibility of an edge absent between two nodes, but for our purposes we are considering a mesh network.
- To automize the algorithm, all input will be provided prior before execution.
- Time taken by the algorithm is calculated using time complexity equation.

- The value of *k*, as we will be using a mesh network, we will assume to be a value in between n/4 to n/2, where n is the number of nodes.

3.2 Implementation Environment:

The minimum implementation environment required is described as follows:

- Processing Power: 1.2 Gigahertz.
- Memory: 1 Gigabytes.
- Secondary Storage: N.A

The algorithm is developed in C language using CodeBlocks IDE.

3.3 Results:

The result provided here is only for the example explained above. The results for higher node numbers cannot be provided as it would be an exhaustive process to analyse it manually.

Furthermore, the time taken by the algorithm to implement for various number of nodes is shown in the table below:

Serial No.	Number of Nodes	Time Taken (in Seconds)
1.	6	0.14
2.	10	0.61
3.	25	8.74
4.	50	66.12
5.	100	517.23
6.	200	4071.58

From the above table we observe that as the number of nodes increase, the time taken for execution also increases. The time complexity value for the algorithm we calculated, i.e. $T= v^2 + v*e$, where v is the number of nodes and e is the number of edges.

We have considered a mesh network. So, for the mesh network the number of edges are $N*(N-1)/2$, where N is the number of nodes. In our time complexity equation if we insert the above equation value for number of edges we get the following time complexity equation:

$T= v^2 + v*e$
$T= v^2 + v*(v*(v-1))/2$
$T= v^2 + (v^3- v^2)/2$

If we calculate time using the above equation and compare it to the time obtained after implementation, we can check the error rate.

Serial No.	Number of Nodes	Obtained Time Value (in Seconds)	Calculated time value (in Seconds)	Error Rate (ΔE)= Obtained Value - Calculated Value	%Error= (ΔE*100)/ (Calculated Value)
1.	6	0.14	0.12	0.02	16.67
2.	10	0.61	0.55	0.06	10.90
3.	25	8.74	8.13	0.61	7.50
4.	50	66.12	63.75	2.37	3.71
5.	100	517.23	505	12.23	2.42
6.	200	4071.58	4020	51.58	1.28

Table 1: Comparison Table

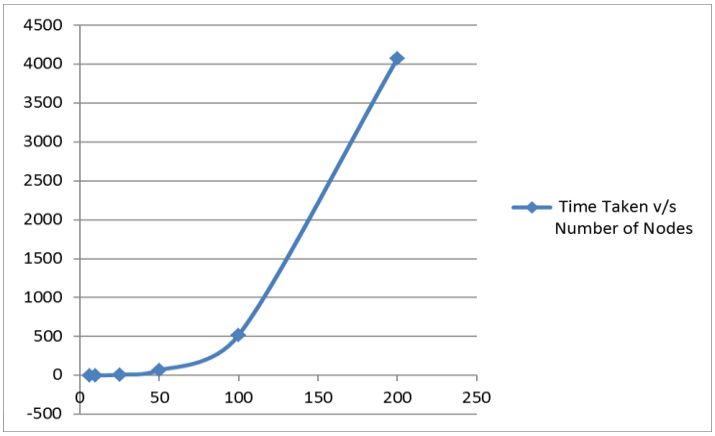


Figure 1: Time Taken v/s Number of Nodes

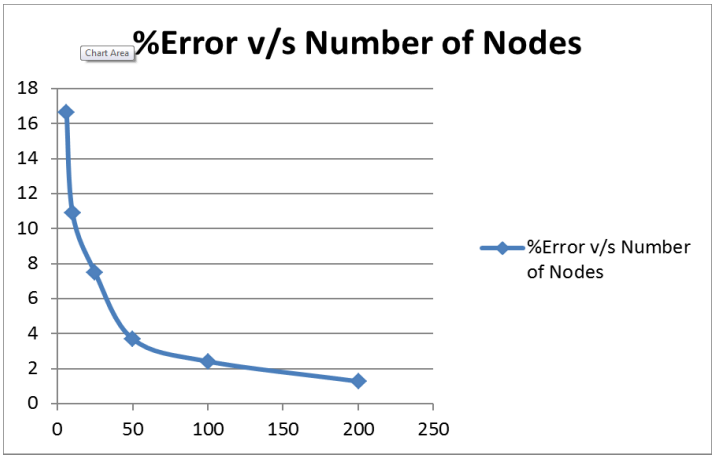


Figure 2: %Error v/s Number Of Nodes

IV. CONCLUSION

The degree constrained minimum spanning tree problem is a NP-Hard problem. Based on my studies we found out those genetic algorithms a part of the evolutionary computing has grown out to be a profound way to solve such problems. The Exon shuffling genetic algorithm provides a very natural way to efficiently solve the complexity problems of bin packing problems and multi knapsack problems. The exon shuffling genetic algorithm can also be used to solve the problem of the degree-constrained minimum spanning tree.Various constraints related to graph have been considered and an algorithm has been prepared for the implementation in the near future. As for the results obtained, it can be said that for more number of nodes the percentage error will be less, thus more accurate. The hardness of the problem, if compared to the fact that a hard problem has exponential or factorial component, the proposed algorithm has polynomial time complexity, thus comparatively reducing the hardness of the problem.

REFERENCES

[1] Degree-constrained spanning tree. (n.d.). Retrieved January 15, 2016, from https://en.wikipedia.org/wiki/Degree-constrained_spanning_tree

[2] Bui, T. N., &Zrncic, C. M. (2006, July). An ant-based algorithm for finding degree-constrained minimum spanning tree. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation* (pp. 11-18). ACM.

[3] Rohlfshagen, P., &Bullinaria, J. A. (2010). Nature inspired genetic algorithms for hard packing problems. *Annals of Operations Research*,*179*(1), 393-419.

