Journal of Computational Intelligence in Bioinformatics
ISSN 0973-385X Volume 5, Number 1 (2012) pp. 45-56

http://www.ripublication.com/jcib.htm

An Influence of Chromosomes Population in Degree Constraint Minimal Spanning Tree Problem using Evolutionary Approach Genetic Algorithm

[1]Dr. Anand Kumar and [2]Dr. N.N. Jani

[1]Department of Master of Computer Applications
M.S. Engineering College, Bangalore, India
E-mail: kumaranandkumar@gmail.com
[2]Faculty of Computer Studies
Kadi Sarva Vishwavidyalya, Gandhinagar, India
E-mail: drnnjaneecsd@gmail.com

Abstract

This paper presents an influence of number of chromosomes population in degree constarint minimal spanning tree problem using genetic algorithm. In this research work, influence of chromosomes are studied and shown that how effectively it works in genetic algorithm. So far very few researchers have studied about the influence of chromosomes population and that is also for small network size. This study has been done for small to large size network and very impressive result has been derived.

Keywords: Genetic Algorithm, Chromosomes, Degree Constrained Minimum Spanning Tree.

Introduction

Chromosomes play an important role in genetic algorithms paradigm. The concept is simple and straight forward where result is derived from the set of chromosomes also called population. In the course of execution, there are two options before the researcher: First to go for maximum number of population and then send these fittest populations after applying degree constraint to next generation and so on. Second option is to start with the minimum number of population but apply maximum number of generation. In this research work, both the options have been executed for small to large set of network for DCMST problem and it is concluded that maximum number of population always works better than more and more generation. The performance of GA depends to a great extent, on the maximum number of chromosomes.

Degree Constraint Minimal Spanning Tree

Degree-constrained minimum spanning tree (DCMST) is an NP hard problem[6] and so far no efficient method has been found[1]. Genetic Algorithm has been always an alternative solution for such type of NP hard problem. The DCMST problem can be defined as

$$1 \leq d(v_i) \leq n - 1 \qquad (1)$$

Where n is the total no of vertex and d is the degree denoted by $d(v_i)$ of a node $i(i = 1, ...n)$. The degree is the number of incident edges, and the degree of a graph is the maximum degree of its nodes. The degree constrained MST problem is to determine a spanning tree of the minimum total edge cost and degree no more than a given value k. Then each node of a network must not be connected with k other nodes. Researchers have attempted [3] for small size of network. In this research work small to large size of network is considered which is one of the main part of this work.

Genetic Algorithm Approach to Find Degree Constraint Minimal Spanning Tree

Genetic Algorithm is a natural evolution process which provides multiple solution[4][5] at a time. This work is the extension work [2] and carry special case of number of chromosomes. In this research work, the work flow of genetic algorithm is defined as following:

Work flow of GA

1. Initialization of parent population.
2. Evaluation
    a. Self loop check
    b. Isolated node or edge check
    c. Cycle check
    d. Store the best result

3. Selection of child population
4. Apply Crossover/ Recombination
5. Evaluation
6. Replace the result if it is better than previously stored.
7. Apply Mutation
8. Evaluation
9. Replace the result if it is better than previously stored.
10. Go to step 3 until termination criteria satisfies

The Degree Constrained Minimum Spanning Tree is considered as a non directional graph and represented with the help of adjacency matrix. Parent population in the form of chromosome is generated randomly according to the size of network. Number of gene in a chromosome is equal to number of node in a network. The total number of chromosome may vary and it is based on user input. Here a chromosome is generated for a 10 node network. The association between nodes is considered between positions to position.

node	1	2	3	4	5	6	7	8	9	10

chromosome	8	6	3	1	8	9	4	2	7	1

The logic behind association is that, the node [1] is connected with node 2; node [2] is connected with 10 and so on.

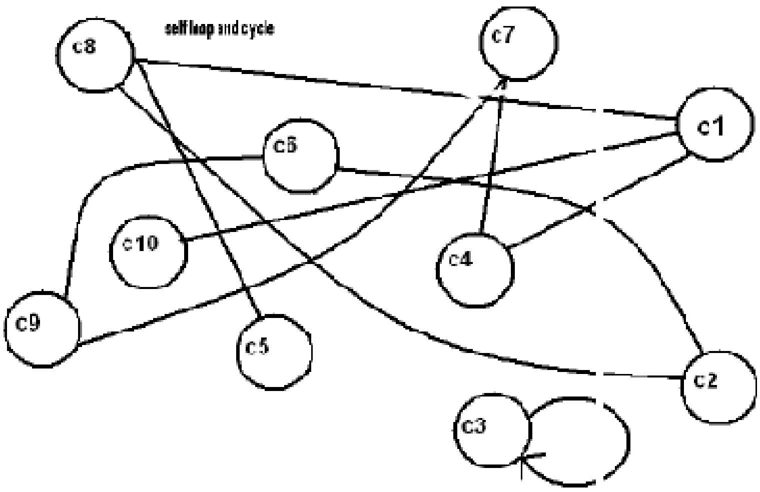

Figure 1

From Figure-1 it is clear that this is not a spanning tree because of the self loop and circle. Similarly with some other randomly generated chromosome, some other problems have been observed as an isolated edge and multiple isolated edges. . By observing these problems it has been concluded that there are three main reasons for illegal chromosome:

i. Self loop
ii. Cycle and
iii. Isolated node or edge.

Accordingly fitness functions [2] are developed and applied.

Chromosomes

Chromosome is the first step of genetic algorithm. It can be represented in the form of digits or alphabets, depending on the requirement of the problem. In this research work, chromosomes are represented in the form of digits. Chromosomes are randomly generated and stored in the file. The main objective of this research work is to see the influence of chromosomes during the calculation of the distance of DCMST problem. The entire experiment is classified in to two parts: population variation with fixed generation and fixed population with generation variation. For both kind of approach, same network is considered and at last cost is compared for both the approaches. Network is considered as a weighted graph where weight represents the distance between two nodes or vertex and the entire network is represented with the help of adjacency matrix and generated and stored in a file..

Experimental Result

The experiment is done in MATLAB R2008a version 7.6.0.324.

Experiment 1

Population size	:	10 to 2000
No of Generations	:	100
Selection	:	fittest chromosomes first
Mutation	:	unfit chromosomes to mutate

Network Size: 10

Table 1

Population	Distance of DCMST
10	330
20	308
40	298
60	273
80	250
100	239
200	256
400	245
600	227
800	236
1000	208

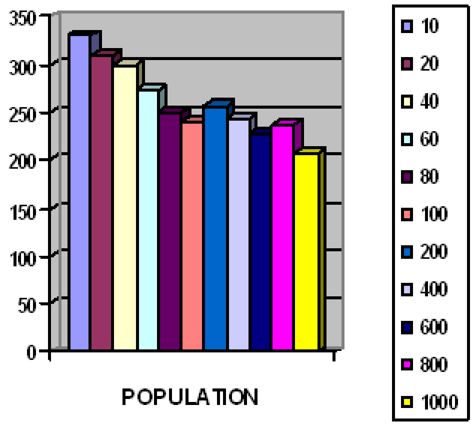

Figure 1

Network Size: 50

Table 2

Population	Distance of DCMST
10	0
20	0
40	1908
60	1794
80	1758
100	1818
200	1659
400	1666
600	1653
800	1649
1000	1627

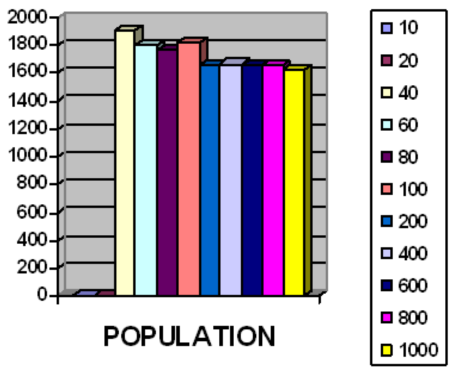

Figure 2

Network Size: 100

Table 3

Population	Distance of DCMST
10	0
20	0
40	0
60	3922
80	3499
100	3669
200	3467
400	3329
600	3278
800	3492
1000	3405

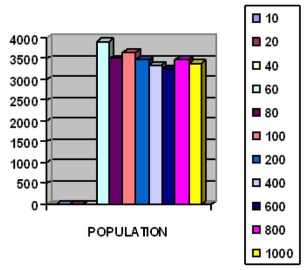

Figure 3

Network Size: 200

Table 4

Population	Distance of DCMST
10	0
20	0
40	0
60	7569
80	0
100	0
200	7565
400	7435
600	7277
800	7371
1000	7282

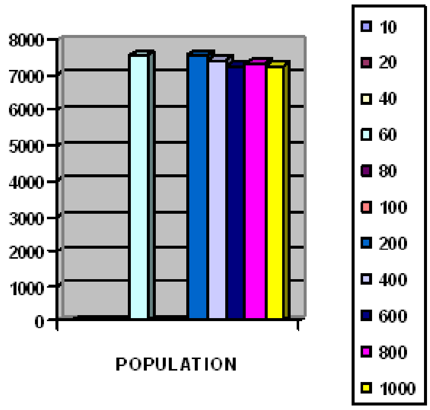

Figure 4

Network Size: 400

Table 5

Population	Distance of DCMST
10	0
20	0
40	0
60	0
80	15490
100	16112
200	15850
400	15265
600	15104
800	14836
1000	15114

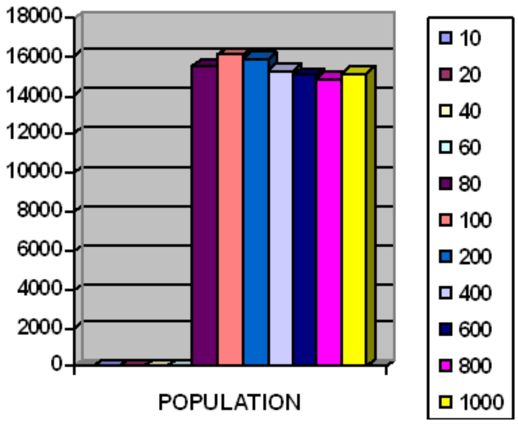

Figure 5

Experiment 2

Population size	:	100
No of Generations	:	10 to 1000
Selection	:	fittest chromosomes first
Mutation	:	unfit chromosomes to mutate

Network Size: 10

Table 6

Generations	Distance of DCMST
10	243
20	243
40	243
60	243
80	239
100	239
200	239
400	239
600	239
800	239
1000	239

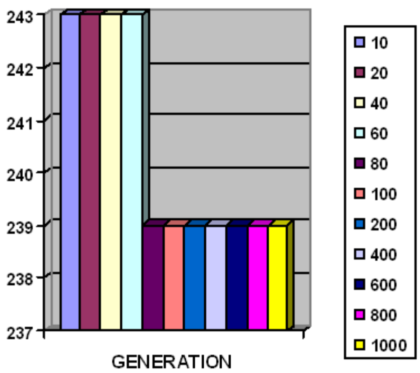

Figure 6

Network Size: 50

Table 7

Generations	Distance of DCMST
10	1836
20	1818
40	1818
60	1818
80	1818

100	1818
200	1818
400	1818
600	1818
800	1818
1000	1818

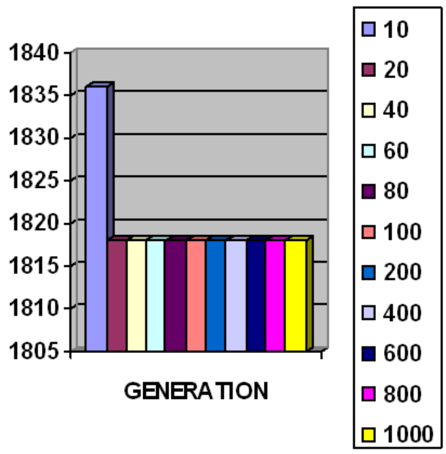

Figure 7

Network Size: 100

Table 8

Generations	Distance of DCMST
10	3681
20	3673
40	3669
60	3669
80	3669
100	3669
200	3669
400	3669
600	3669
800	3669
1000	3669

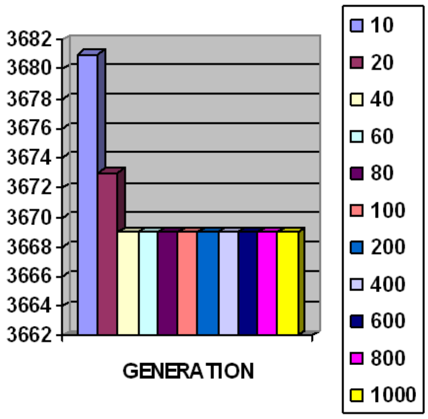

Figure 8

Network Size: 400

Table 9

Generations	Distance of DCMST
10	16112
20	16112
40	16112
60	16112
80	16112
100	16112
200	16112
400	16112
600	16112
800	16112
1000	16112

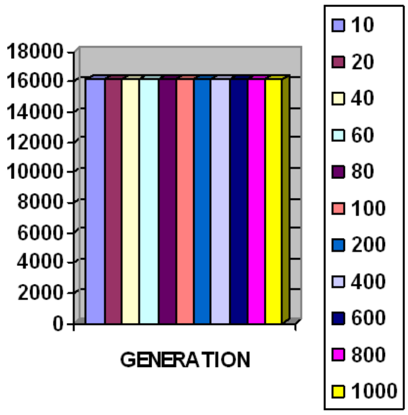

Figure 9

Experiment 3
Population size : 400
No of Generations : 10 to 1000
Selection : fittest chromosomes first
Mutation : unfit chromosomes to mutate

Table 10

Generations	Network Size				
	10	50	100	200	400
10	258	1755	3557	7464	15265
20	258	1749	3329	7464	15265
40	245	1667	3329	7435	15265
60	245	1667	3329	7435	15265
80	245	1667	3329	7435	15265
100	233	1666	3329	7435	15265
200	233	1658	3329	7435	15265
400	233	1658	3329	7435	15265
600	233	1658	3329	7435	15265
800	233	1658	3329	7435	15265
1000	233	1658	3329	7435	15265

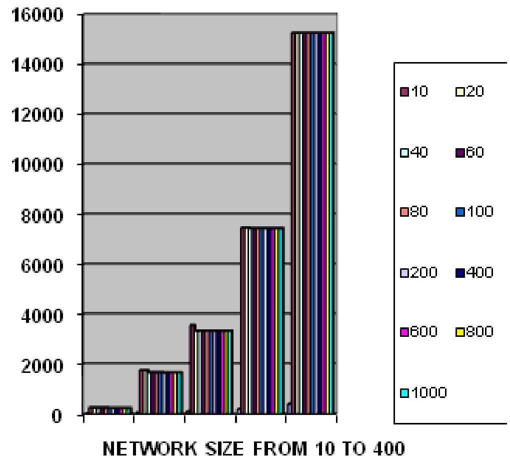

Figure 10

Conclusions

From the above three experiments, following results have been concluded:

i. Experiment 1 provides the better result. With the maximum number of population, there are maximum chance to get the better result. From figure 1 to figure 5, it is concluded that results are almost improved with the population variation for small to large size network. Average reults are improving towards minimum to maximum number of population.

ii. Experiment 2 provides the constant result. Initialy results are improving but in the later generations, results become constant with no improvement. One main reason is that all the chromosomes become same and that is why genetic operators are ineffective.
iii. Experiment 3 also has the same result as experiment 2, even thou populations are four times increased (400).
iv. The effectiveness of the methodology however can be increased by applying the various genetic operators with variations of network size as the densely connected locations.

Acknowledgement
Our thanks to the reviewers for their cogent and insightful comments

References

[1] Narsingh Deo, 2000. Graph Theory with Applications to Engineering and Computer science: (PHI)
[2] Anand Kumar and Dr. N. N. Jani, A Novel Genetic Algorithm approach for Network Design wit Robust Fitness Function International Journal of Computer Theory and Engineering, Vol. 2, No. 3, June, 20101793-8201
[3] Hsinghua Chou, G. Premkumar, and Chao-Hsien Chu, Genetic Algorithms for Communications Network Design—An Empirical Study of the Factors that Influence Performance, IEEE Transactions On Evolutionary Computation, Vol. 5, No. 3, June 2001
[4] Melanie M. (1998). An Introduction to genetic Algorithm (PHI) ISBN 81-203-1385-5
[5] Michael D. Vose. 1999. The simple genetic algorithm : (PHI) ISBN 61-203-2459-5
[6] Lixia Hanr† and Yuping Wang, A Novel Genetic Algorithm for Degree-Constrained Minimum Spanning Tree Problem, IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.7A, July 2006