International Journal of Information Technology and Knowledge Management
July-December 2010, Volume 2, No. 2, pp. 605-611

Genetic Algorithm for Network Design Problem-An Empirical Study of Crossover Operator with Generation and Population Variation

Anand Kumar & N. N. Jani

This paper presents an influence of genetic operators (crossover) in genetic algorithm for network design problem. It also describes the performance variation with the number of generation and number of chromosomes for various different sizes of networks. A network design problem for this paper falls under the network topology category which is a degree constrained minimum spanning tree with various types of constraint demanded by the current requirement which makes it NP-hard problem. Genetic operators play an important key role in genetic algorithm approach. Since many researchers have tried to solve this problem for small to mid size, we have explored the use of genetic algorithm with various operators with modification but without changing the nature of genetic algorithm. Various crossover and mutation operators have been developed here as per the requirement of the problem and applied with the various size of network. In this paper we have tried to show that how genetic operators affects the performance of genetic algorithm and also shown that GA is an alternative solution for this NP-hard problem.
Keywords: Genetic Algorithm, Network Design, Crossover, Mutation, Minimum Spanning Tree.

1. Introduction

In Genetic Algorithm after encoding and selection the third decision to make in implementing is to use genetic operators. Genetic operators play an important key role to achieve the optimal result. It could be said that the main distinguish feature of genetic algorithm is genetic operator. This paper is the extension of work [10] and describes the importance and influence of genetic operators for network design problem. The network design problem is a NP-hard problem [2] [6] and genetic algorithm approach to design the network is one of the ultimate solutions because traditional heuristics has the limited success. Many researchers have tried to solve the problem [1] [3] [4]. However there is very limited research on using GA for network design with the influence of genetic operators. This paper presents the different variation in crossover and mutation operator with its impact on network size. It also describes the relationship between number of generation and number of chromosomes generated in genetic algorithm approach. Still no researchers have tried to explore this field of genetic algorithm approach.

Network Design

In this paper network design is considered as network topology which is a degree constraint spanning tree consists of various nodes considered as vertex. A tree is a connected graph containing no cycles.

A spanning tree of a connected, undirected graph is just a sub graph that contains all the vertices and is a tree. A graph may have many spanning trees. A minimum spanning tree (MST) is a spanning tree with weight less than or equal to the weight of every other spanning tree A MSTP can be defined as:

For an Undirected Connected Graphs

G = (V, E) a subgraph T = (V, E) of G is a spanning tree of G if T is a tree. Where V is the set of vertices and E is the set of possible edges between pair of vertices, and for each edge (u, v) $\in$ E. There is a weight w(u, v) specifying the distance to connect u and v. The spanning tree T connects all the vertices and the total weight of the tree should be:

$$w(T) = \sum_{(u,v)\in T} w(u,v) \qquad (1)$$

The MST algorithm may occasionally generate a minimal spanning tree where all the links connect to one or two nodes. This solution, although optimal, may be highly vulnerable to failure due to over reliance on a few nodes. Furthermore, the technology to connect many links to a node may not be available or may be too expensive. Hence, it may be necessary to limit the number of links connecting to a node. Alternatively, from reliability perspective it is desirable to have more than one link connect to a node so that alternative routes can be selected in the case of a node or link failure [5]. DCMST was specifically developed as a special case of MST with additional constraints to improve

[1]Department of Master of Computer Applications AMC Engineering College, Bangalore
[2]Faculty of Computer Studies, Kadi Sarva Vishwavidyalya
Email: [1]kumaranandkumar@gmail.com,
[2]Gandhinagardrnnjanicsd@gmail.com

the reliability of the network and rerouting of traffic in the case of node failures. This is the extended version of minimum spanning tree problem where an extra constraint is added with each vertex. When this extra constraint is added this problem becomes NP-hard [6]. This constraint is usually motivated by the need to impose a limit on the number of ports in each node. In a shortest spanning tree resulting from the preceding construction, a vertex v_i can end up with any degree; that is

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

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.

$$d(v_i) < k, \ i = 1, ...n.$$

Work Flow of GA

1. Initialisation of parent population;
2. Evaluation;
   a) Self loop check,
   b) Isolated node or edge check,
   c) Cycle check,
   d) Path constraint,
   e) Degree constraint,
   f) 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.

2. GENETIC ALGORITHM APPROACH TO SOLVE THE PROBLEM

The Network design problem is considered as a unidirectional 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	2	10	4	9	6	7	5	9	8	3

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

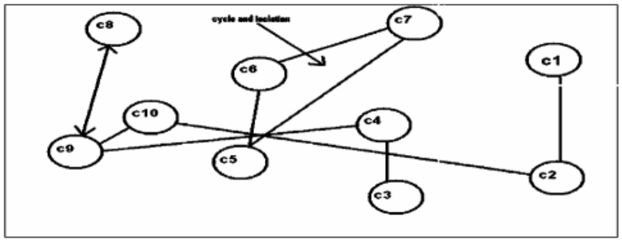

Fig. 1

From fig-1 it is clear that this is not a spanning tree because of the isolated circle. Similarly with some other randomly generated chromosome, some other problems have been observed.

By observing these problems it has been concluded that there are five main reasons for illegal chromosome:

a) Self loop;
b) Isolated node or edge;
c) Cycle;
d) Path constraint;
e) Degree constraint.

Evaluation

By observing these problems, fitness functions have been developed to evaluate these chromosomes. On the basis of these fitness functions, fitness points are given to the chromosomes and on the basis of these fitness points chromosomes are selected as a child population for next generation. Since the paper is based mainly based on genetic operators so details of these fitness function is not given here but it is available in Anand Kumar and Dr. N.N.Jani [10].

Selection

The purpose of selection is to emphasize the fitter individuals in the population in hopes that their offspring will in turn have even higher fitness. In this research Roulette wheel Selection function is used for the selection of child population.

3. GENETIC OPERATORS

This is the main part of this research paper. After the population generation, evaluation and selection of the child population this is phase of genetic algorithm which describes that how to rebuild the population with the existing resource. Genetic operators plays important key role in genetic algorithm approach. Genetic operators: must be designed based on the problem. There are mainly two types of genetic operators but variation are available and this is the subject of research. The main two types of operators are crossover and mutation:

Crossover

It could be said that the main distinguishing feature of genetic algorithm is the use of crossover. This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. Single point crossover is the simplest form. The idea here is, of course to recombine building blocks on different strings. Single-point crossover has some shortcomings, though. For one thing it cannot combine all possible schemas [7]. To reduce positional bias two- point crossover is applied in which two positions are chosen at random and the segments between them are exchanged. Two-point crossover is less likely to disrupt schemas with large defining lengths and can combine more schemas than single point crossover. In addition the segments that are exchanged do not necessarily contain the endpoints of the string. In this research work three different crossover functions have been developed.

A. Single_point _crossover

This crossover function crossovers each pair of chromosome on a different single point. First pair of chromosome will be crossover on first column from left, similarly next second pair of chromosome is crossover on second point and so on.

```
function [chromosomes] = single_point_crossover
(chromosomes)
a=size(chromosomes);
t=1;i=1;
while( i<row)
 for j=1:t
 temp = chromosomes(i,j);
 chromosomes(i,j) = chromosomes((i+1),j);
 chromosomes((i+1),j)=temp;
 end
 i=i+2;
 t=t+1;
 if(t>col)
 t=1;
 end
end
```

B. Double_fixed-point_crossover

This function crossovers the chromosome on fixed two different points for each pair of the chromosome.

```
function [chromosomes] =
double_fixed-point_crossover12(chromosomes)
disp(chromosomes);
a=size(chromosomes);
t=1;i=1;
if(p==0)
 p=1;
end
 if(q==0)
 q=1;
 end
if(p>q)
 p1=q;
 p2=p;
else
 p1=p;
 p2=q;
end
while( i<row)
 for j=p1:p2
 temp = chromosomes(i,j);
 chromosomes(i,j) = chromosomes((i+1),j);
 chromosomes((i+1),j)=temp;
 end
 i=i+2;
 t=t+1;
 if(t>col)
 t=1;
 end
end
end
```

C. Double_vary-point_crossover

This function crossovers the chromosome on different two different points for each pair of the chromosome. This is the variation of Double_fixed-point_crossover.

```
function [chromosomes] =
double_vary-point_crossover13(chromosomes)
i=1;
while( i<row)
 p=round(rand()*col);
 if(p==0)
 p=1;
 end
q=round(rand()*col);
 if(q==0)
 q=1;
 end
if(p>q)
 p1=q;
 p2=p;
else
 p1=p;
 p2=q;
end
 for j=p1:p2
 temp = chromosomes(i,j);
 chromosomes(i,j) = chromosomes((i+1),j);
 chromosomes((i+1),j)=temp;
 end
 disp(chromosomes);
 i=i+2;
end
```

Mutation

This operator randomly flips some of the bits in a chromosome. Mutation can occur at each bit position with some probability, usually, very small. This mutation operator mutates only those chromosomes which does not have the full fitness. The logic is quite simple that find the chromosome and change its value with its position. If first chromosome is selected then its first place will be replaced by maximum no (say N). If the second chromosome is selected then second place will be selected by (N-1) and so on.

```
function [mutated_chromosome] = mutation1(new_chromosome)

a=size(new_chromosome);

k=1;

for i=1:row

 if(new_chromosome(i,a(2)) ~= (a(2)+3))

 new_chromosome(i,i) = (a(2)-i);

 end

end

for i=1:row

 for j=1:a(2)-1

 mutated_chromosome(i,j) = new_chromosome(i,j);

 end

end
```

4. Experimental Result

Ten different network of size 10,60……100 have been used. Total number of chromosomes and total number of generations for each network is 100. The experiment is done in MATLAB R2008a version 7.6.0.324. All the crossover is experimented with various size of network and population –generation combination. Following tables and figures display the result:

Table 1
Minimum Cost of Network for Various Crossover Operators-network Size -10

Population	generation	Single Point crossover	Double_fixed point crossover	Double vary-point crossover
10	100	307	297	315
20	100	257	302	277
30	100	286	260	252
40	100	286	308	276
50	100	269	274	233
60	100	251	221	263
70	100	247	286	251
80	100	271	240	278
90	100	262	254	245
100	100	231	268	207
200	100	237	245	254
300	100	242	258	252
400	100	248	229	226
500	100	249	225	195
1000	100	210	197	237

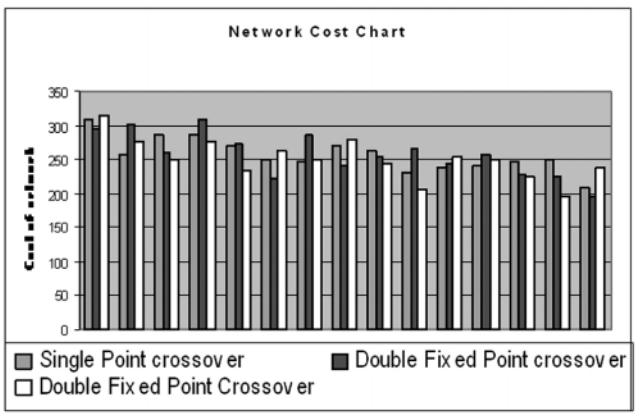

Fig. 2

Table 2
Minimum Cost of Network for Various Crossover Operators-Network Size-10

Population	generation	Single Point crossover	Double_fixed point crossover	Double vary-point crossover
100	10	317	298	291
100	20	248	265	263
100	30	249	280	261
100	40	256	251	284
100	50	255	245	271
100	60	222	260	267
100	70	220	223	258
100	80	265	229	225
100	90	242	274	228
100	100	241	226	255
100	200	237	246	262
100	300	234	266	209
100	400	250	253	241
100	500	248	235	270
100	600	232	255	233
100	700	208	248	224
100	800	240	251	238
100	900	265	213	240
100	1000	254	272	256

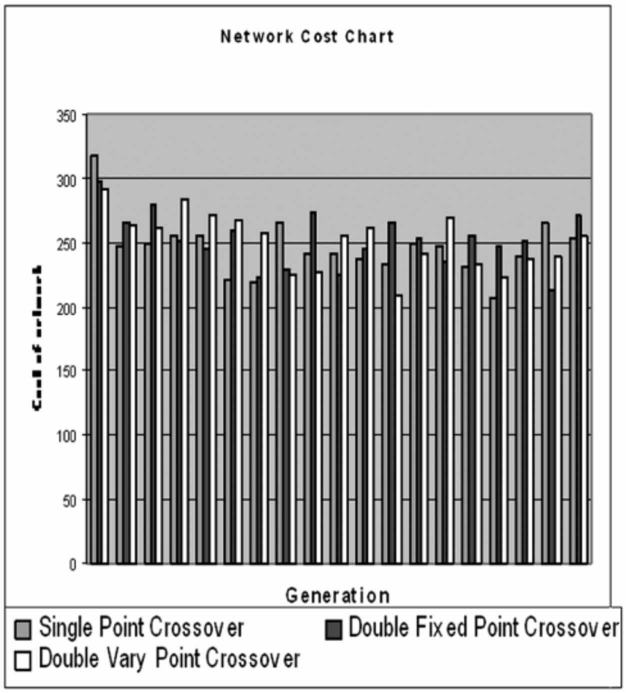

Fig. 3

Table 3
Minimum Cost of Network for Various Crossover Operators- Network Size -60

Population	generation	Single Point crossover	Double_fixed point crossover	Double vary-point crossover
20	100	0	2233	0
40	100	2457	2425	0
60	100	0	2285	2211
80	100	2246	2253	2166
100	100	2169	2166	2273
200	100	2161	2091	2142
300	100	2014	2084	2010
400	100	2053	2130	1933
500	100	2047	2066	1954
600	100	2031	2177	2065
700	100	2041	2047	2040
800	100	2013	1905	1983
900	100	2004	1983	2105
1000	100	1962	2032	2041

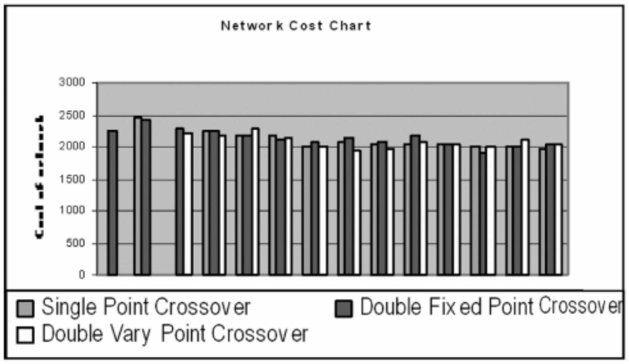

Fig. 4

Table 4
Minimum Cost of Network for Various Crossover Operators-Network Size-60

Population	generation	Single Point crossover	Double_fixed point crossover	Double vary-point crossover
100	10	0	0	2161
100	20	2133	2364	2386
100	30	2179	2212	2190
100	40	2189	2178	2170
100	50	2203	2201	2284
100	60	2061	2074	2187
100	70	2276	2216	2209

Contd...

Contd...

100	80	2248	2256	2151
100	90	2084	2013	2059
100	100	2254	2145	2092
100	200	2226	2218	2184
100	300	2081	2111	2234
100	400	2212	2152	2381
100	500	2076	2083	2245
100	600	2228	2151	2121
100	700	2140	2106	2139
100	800	2123	2180	2107
100	900	2134	2154	2144
100	1000	0	2238	2064

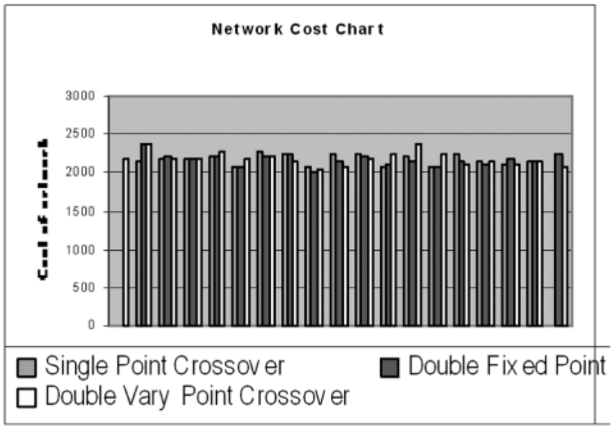

Fig. 5

Table 5
Minimum Cost of Network for Various Crossover Operators-Network Size-100

Population	generation	Single Point crossover	Double_fixed point crossover	Double vary-point crossover
100	10	0	0	0
100	20	0	0	0
100	30	0	0	0
100	40	0	3789	3256
100	50	0	0	0
100	60	0	0	0
100	70	0	0	0
100	80	0	0	0
100	90	3883	0	0
100	100	0	0	0
100	200	0	0	0
100	300	0	0	0
100	400	0	0	3456
100	500	0	0	0
100	600	0	0	0
100	700	0	0	0
100	800	4356	0	4123
100	900	0	0	0
100	1000	0	3245	0

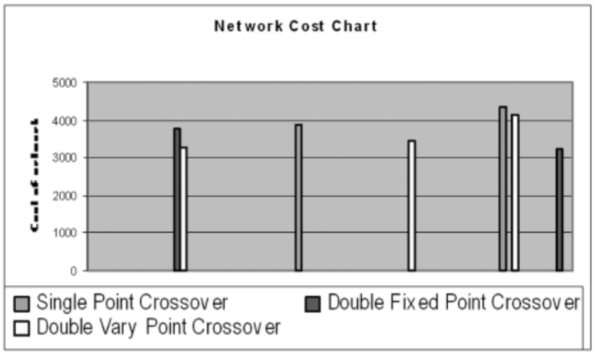

Fig. 6

Table. 6
Minimum Cost of Network for Various Crossover Operators- Network Size -100

Population	generation	Single Point crossover	Double_fixed point crossover	Double vary-point crossover
50	100	0	0	0
100	100	0	0	3768
200	100	0	0	3478
300	100	3877	3855	3659
400	100	3666	3543	3624
500	100	3767	3874	3693
600	100	4013	3669	3547
700	100	3886	3335	3699
800	100	3692	3998	3678
900	100	3673	3321	3378
1000	100	3819	3330	3221

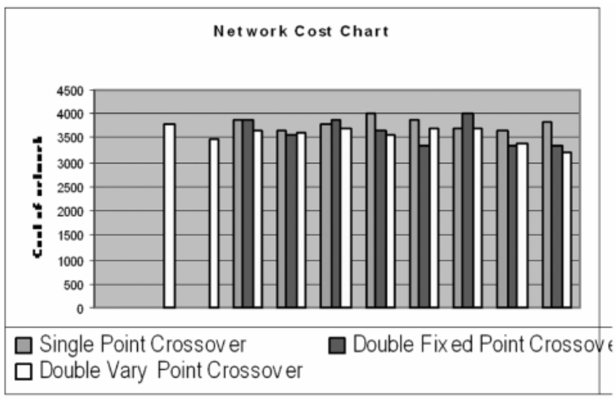

Fig. 7

5. CONCLUSIONS

From the above experimental result of Table-1 to Table-6 and the figure1 to figure7, following conclusions have been made.

- For the smaller size network (Network Size 10), there is possibility of better result with the maximum generation and maximum population and double fixed point crossover.
- For the medium size network (Network Size 60), there is possibility of better result with the maximum generation and all the crossover function has almost same impact.
- For the larger size network (Network Size 100), there is possibility of better result with only maximum population and double point crossover (fixed and varying)

ACKNOWLEDGEMENTS

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

REFERENCES

[1] M. Gruber and G. Raidl. "A New 0.1 ILP Approach for the Bounded Diameter Minimum Spanning Tree Problem". In L. Gouveia and C. Mour.ao, Editors, Proc. of the Int. Network Optimization Conference, 1, Pages 178-85, Lisbon, Portugal, 2005.

[2] Ellis Horwitz, Sartaj Sahni and Sanguthevar Rajasekaran, Computer Algorithms, University Press, 2007.

[3] Rajeev Kumar and Nilanjan Banerjee, "Multicriteria Network Design Using Evolutionary Algorithm", GECCO 2003, LNCS 2724, pp. 2179–2190, 2003, Springer-Verlag Berlin Heidelberg 2003.

[4] Lixiar Hanr and Yuping Wang, "A Novel Genetic Algorithm for Degree Constrained Minimum Spanning Tree Problem", International Journal of Computer Science and Network Security, 6, No. 7A, July 2006.

[5] F. Harary and J. P. Hayes, "Node Fault Tolerance in Graphs," Networks, 27, No. 1, pp. 19–23, 1996.

[6] S.K. Basu, 2005. "Design Methods and Analysis of Algorithms (PHI)" ISBN : 81-203-2637-7.

[7] Melanie M. (1998). "An Introduction to Genetic Algorithm (PHI)" ISBN 81-203-1385-5.

[8] Michael D. Vose. 1999. "The Simple Genetic Algorithm: (PHI) ISBN 61-203-2459-5.

[9] Narsingh Deo, 2000. Graph Theory with Applications to Engineering and Computer Science: (PHI).

[10] Anand Kumar, Dr. N.N.Jani, Proceeding of the International Conference on Mathematics and Computer Science ICMCS FEB 2010, ISBN: 978-81-908234-2-5 Chennai.