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:

\[
 w(T) = \sum\limits_{(u,v)\in T} w(u,v)
\]  

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...
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 \( (i = 1, \ldots, n) \). 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 \( k \).

\[ d(v_i) < k, \: i = 1, \ldots, n \]

**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](image)

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.

```matlab
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;
end
```

B. Double_fixed-point_crossover

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

```matlab
function [chromosomes] = double_fixed-point_crossover(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;
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.

```matlab
function [chromosomes] = double_vary-point_crossover(chromosomes)
disp(chromosomes);
a=size(chromosomes);
t=1;i=1;
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
```
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

end

mutation1(chromosomes)
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>
<thead>
<tr>
<th>Population</th>
<th>Single Point crossover</th>
<th>Double_fixed point crossover</th>
<th>Double vary-point crossover</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>307</td>
<td>297</td>
<td>315</td>
</tr>
<tr>
<td>20</td>
<td>257</td>
<td>302</td>
<td>277</td>
</tr>
<tr>
<td>30</td>
<td>286</td>
<td>260</td>
<td>252</td>
</tr>
<tr>
<td>40</td>
<td>286</td>
<td>308</td>
<td>276</td>
</tr>
<tr>
<td>50</td>
<td>269</td>
<td>274</td>
<td>233</td>
</tr>
<tr>
<td>60</td>
<td>251</td>
<td>221</td>
<td>263</td>
</tr>
<tr>
<td>70</td>
<td>247</td>
<td>286</td>
<td>251</td>
</tr>
<tr>
<td>80</td>
<td>271</td>
<td>240</td>
<td>278</td>
</tr>
<tr>
<td>90</td>
<td>262</td>
<td>254</td>
<td>245</td>
</tr>
<tr>
<td>100</td>
<td>231</td>
<td>268</td>
<td>207</td>
</tr>
<tr>
<td>200</td>
<td>237</td>
<td>245</td>
<td>254</td>
</tr>
<tr>
<td>300</td>
<td>242</td>
<td>258</td>
<td>252</td>
</tr>
<tr>
<td>400</td>
<td>248</td>
<td>229</td>
<td>226</td>
</tr>
<tr>
<td>500</td>
<td>249</td>
<td>225</td>
<td>195</td>
</tr>
<tr>
<td>1000</td>
<td>210</td>
<td>197</td>
<td>237</td>
</tr>
</tbody>
</table>

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

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

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

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

Fig. 3
Table 4
Minimum Cost of Network for Various Crossover Operators-Network Size-60

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

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

<table>
<thead>
<tr>
<th>Population generation</th>
<th>Single Point crossover</th>
<th>Double_fixed point crossover</th>
<th>Double vary-point crossover</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>40</td>
<td>0</td>
<td>3789</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>60</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>70</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>0</td>
<td>3883</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>200</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>300</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>400</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>500</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
5. Conclusions

From the above experimental result of Table-1 to Table-6 and the figure 1 to figure 7, 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


