Genetic Algorithm Research Paper

Harmandeep Mangat

I. Introduction

This report will be comparing the results of multiple crossovers in a genetic algorithm. A genetic algorithm is a search technique that is inspired by Charles Darwin's theory of natural evolution, where the fittest individuals are selected for crossover and mutation. There are multiple selections algorithms out there that can be used to select the fittest individuals, but the one that was used for these experiments was tournament selection. For crossover, a technique used for reproduction, the techniques that will be looked at are; Ordered Crossover, Uniform Crossover, and a modified ordered crossover. Whereas for mutation, which entails altering the genes of the children to allow the possible solutions to jump to different parts of the search space, the algorithm that was applied was inverse mutation. For this specific case, multiple experiments with different parameters were used to reconstruct a shredded document to see how they affected the genetic algorithm and which crossover performed better.

II. Background

A. Genetic Algorithm

Create the initial population of size n, where each individual is a chromosome and evaluate each of their fitness using a fitness function that represents your problem. Then in a loop with a termination condition being the generation span, the the initial population will go through a selection algorithms, in this case, tournament selection, before applying crossover, mutation, and elitism.

1) Pseudo Code:

```
Create initial population and evaluate fitness
for i <= generation span
    selection algorithm
    crossover
    mutation
```

B. Initial Population

Randomly select 15 genes, where a gene could be any number between 0 and 14 inclusive, evaluate the fitness of the chromosome and add it to the initial population. Repeat until population size is met.

1) Pseudo Code:

```
for i <= population size
    int[] Chromosome
    for i <= 15
        random number between 0 and 14
        add to chromosome array
    evaluate the fitness of the chromosome
    if (current chromosome < best overall fitness) best
        chromosome = current chromosome
  add chromosome and fitness to the initial population
```

C. Tournament Selection

Randomly select k individual, where k represents the number of individual, usually k is between 2-5 inclusive, $k = 2$ used in these experiments. Then compare their fitness, and the one with the better fitness, in this case, the one with the lower fitness, to move on. Repeat until population size is met.

1) Pseudo Code:

```
for i <= population size
    randomly select 2 individuals
    compare their fitness
    new population = chromosome with the lower -
                     fitness
return new population
```

D. Ordered Crossover

Randomly pick two parents and apply Ordered crossover on them. Which entails, randomly selecting sequential elements from one parent and filling in the rest of the genes from the other parent in order. Do the same for the other parent to create two children for the new generation. Repeat until population size is met.

1) Pseudo Code:

```
for i <=population size
    randomly pick two parents
    use parent1 to get random sequential elements
    place sequential elements in their appropriate -
        location in child1
    using parent2, fill in the remaining elements in order
        in child1, ensuring not duplicates.
    Repeat for parent2
    add child1 and child2 with their respected fitness to
        new generation
    if (fitness of child1 or child2 < best overall fitness)
        best chromosome = either child1 or
            child2
```

i++
return new generation

E. Uniform Crossover

Randomly pick two parents and create a mask, which is an array of 1s and 0s. Using the mask, 1 representing the index of the genes to keep and 0 representing the genes to change, transfer the genes that are kept to the child and fill in the remaining from the other parent, ensuring no duplicates.

1) Pseudo Code:

for $i <=$ population size
randomly pick two parents
create a mask
use mask to find the genes that don't change from parent1
place unchanged genes in child1 in their appropriate location
fill in the remaining genes of child1 using parent2
repeat for parent2
add child1 and child2 with their respected fitness to new generation
if (fitness of child1 or child2 $<$ best overall fitness)
best chromosome = either child1 or child2
i++
return new generation

F. Modified ordered Crossover

This introduces a female mate choice into an ordered crossover. In nature, mating choice is a key element of a mating system, and in most species, the females are the choosers. They can choose certain mates for a numerous reasons, one such being, good genes. In a genetic algorithm, that would be the chromosome with a better fitness. This can be applied to an ordered crossover by randomly picking one chromosome that represents a female, and randomly picking x amount of chromosomes, x = 3 for this particular genetic algorithms, that represent the male population. From the male population, the chromosome with the best genes (fitness) will be chosen to mate (perform a crossover) with the female. The steps to performing the actual crossover after the parent selection will be the same as the ordered crossover.

1) Pseudo Code:

for $i <=$population size
randomly pick one chromosome to represent a female
randomly pick 3 chromosomes that represent the male population
from the male population, pick the chromosome with the best fitness as parent2
use parent1 to get random sequential elements
place sequential elements in their appropriate location in child1
using parent2, fill in the remaining elements in order in child1, ensuring not duplicates.
Repeat for parent2
add child1 and child2 with their respected fitness to new generation
if (fitness of child1 or child2 $<$ best overall fitness)
best chromosome = either child1 or child2
i++
return new generation

G. Inverse Mutation

Randomly select k length of sequential elements and flip the order

1) Pseudo Code:

randomly pick sequential elements
int end = end of sequential elements
for i = beginning of sequential elements to end of sequential elements
swap child[i] with child[end]
end–
if $end <= i$ break
return child

H. Elitism

Replace the chromosome with the worst fitness in the current generation with the chromosome with the best fitness found so far

1) Pseudo Code:

int index of worst chromosome
for i ¡ population size
find index of worst chromosome
replace the worst chromosome with the best chromosome found so far

III. Experimental Setup

Five experiments were performed using three different shredded documents and crossovers. For each document, the genetic algorithm was run five times using the three different crossovers. Once the genetic algorithm is run; enter a file name where all the data would be saved, enter the file name of the shredded document, enter the population size, generation span, crossover rate, mutation rate, and finally, either 1, 2, or 3, for ordered crossover, uniform crossover, or modified ordered crossover respectively.

A. Experiment 1

population size: 100
generation span: 100
Crossover rate: 100
Mutation rate: 0

B. Experiment 2

population size: 100
generation span: 100
crossover rate: 100
mutation rate: 10

C. *Experiment 3*

population size: 100
generation span: 100
crossover rate: 90
mutation rate: 0

D. *Experiment 4*

population size: 100
generation span: 100
crossover rate: 90
mutation rate: 10

E. *Experiment 5*

population size: 100
generation span: 100
crossover rate: 70
mutation rate: 6.67

IV. RESULTS

A. *Experiment 1*

The ordered crossover takes longer to converge (see Fig. 1) but produces the best fitness (see table 1). Whereas the modified ordered crossover takes the least amount of generation to converge (see Fig. 3) but produces a higher fitness compared to ordered crossover (see table 2). As for uniform crossover, its convergence performance was in between ordered crossover and modified ordered crossover (See Fig. 2), while producing the worst overall final fitness. These results hold for both shredded document 2 and 3 (See Fig. 4-9).

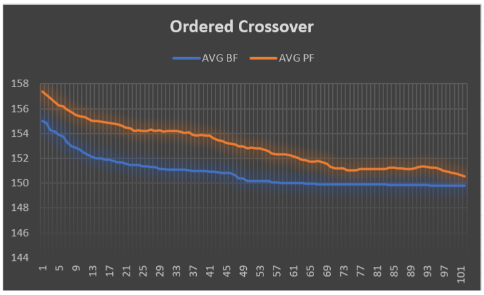

Fig. 1: Ordered Crossover Doc-1

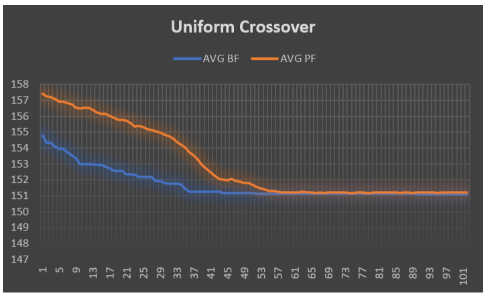

Fig. 2: Uniform Crossover Doc-1

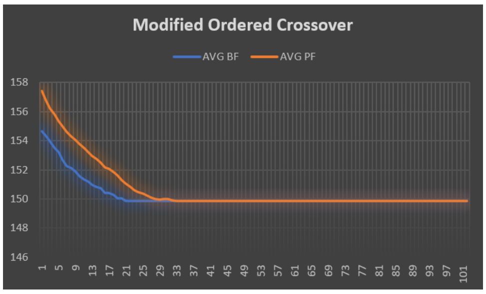

Fig. 3: Modified Ordered Crossover Doc-1

	Average Best	Population Best
Final Fitness	149.7788	150.5488
Min	149.7788	150.5488
Max	155.0224	157.379
Mean	150.8238	153.0009249
Median	150.1849	152.8254931
Standard Deviation	1.236685061	1.759091559

TABLE I: Figure 1 Table

	Average Best	Population Best
Final Fitness	149.8586907	149.8586907
Min	149.8586907	149.8586907
Max	154.6499198	157.4300605
Mean	150.2431004	150.694643
Median	149.8586907	149.8586907
Standard Deviation	1.009188239	1.762738255

TABLE II: Figure 3 Table

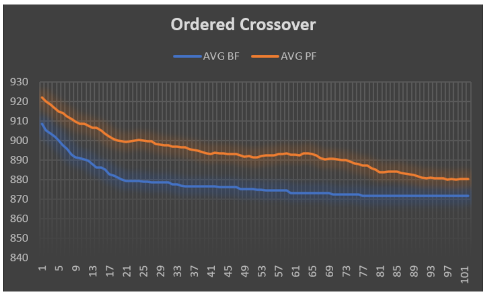

Fig. 4: Ordered Crossover Doc-2

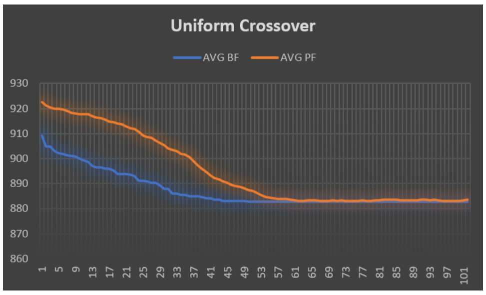

Fig. 5: Uniform Crossover Doc-2

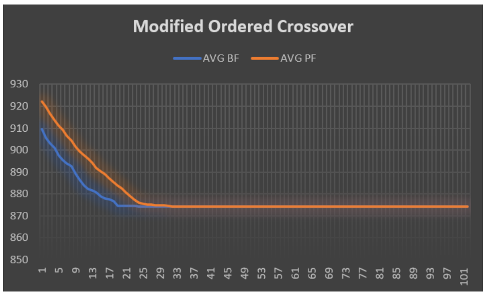

Fig. 6: Modified Ordered Crossover Doc-2

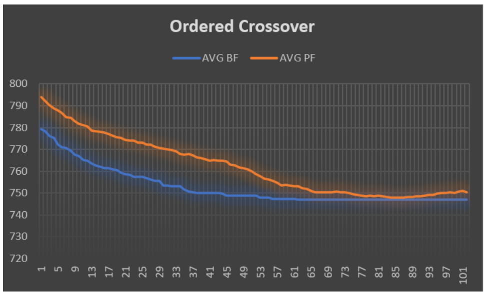

Fig. 7: Ordered Crossover Doc-3

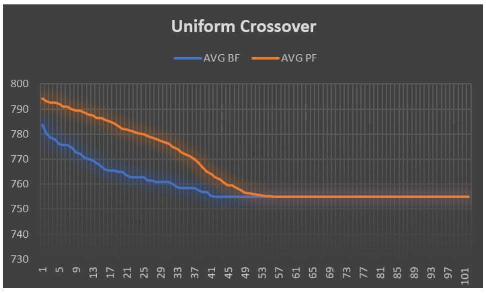

Fig. 8: Uniform Crossover Doc-3

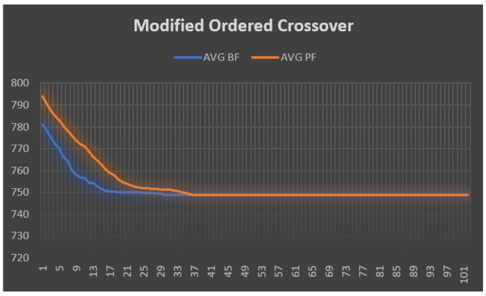

Fig. 9: Modified Ordered Crossover Doc-3

B. Experiment 2

The results were the same as experiment 1, where ordered crossover reaches convergence much later but produces the overall best fitness (see Fig 10 and Table 3). While the modified ordered crossover reaches convergence much earlier (see Fig 12, and table 4).

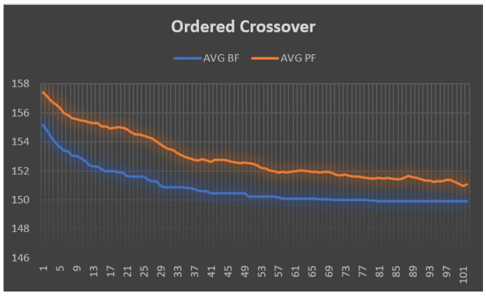

Fig. 10: Ordered Crossover Doc-1

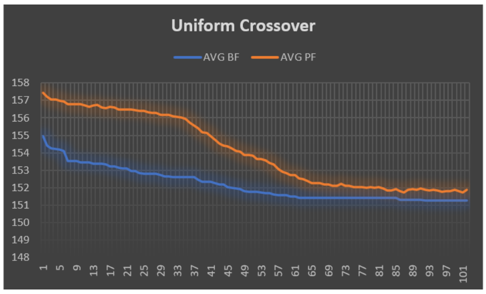

Fig. 11: Uniform Crossover Doc-1

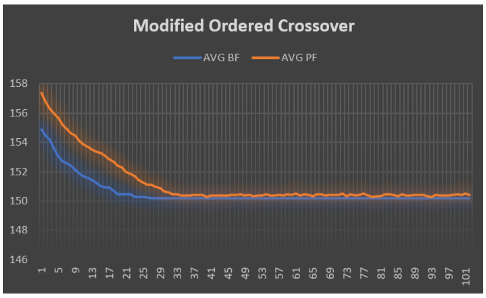

Fig. 12: Modified Ordered Crossover Doc-1

	Average Best	Population Best
Final Fitness	149.9243483	151.0539916
Min	149.9243483	150.9755644
Max	155.1785764	157.4093643
Mean	150.8394594	152.9461097
Median	150.2275993	152.438013
Standard Deviation	1.216658188	1.654982487

TABLE III: Figure 10 Table

	Average Best	Population Best
Final Fitness	150.1724362	150.4277136
Min	150.1724362	150.2848647
Max	154.8806577	157.382474
Mean	150.5611152	151.2550141
Median	150.1724362	150.4265017
Standard Deviation	0.966273466	1.677891888

TABLE IV: Figure 12 Table

C. Experiment 3

Once again the results were similar to the previous experiments. Modified crossover reaches converge faster then the other two (See Fig. 15), but doesn't produce the best fitness, ordered crossover does (See table 15 and 13).

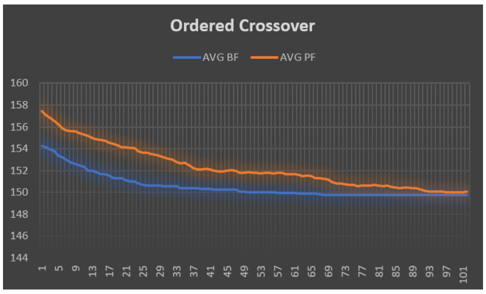

Fig. 13: Ordered Crossover Doc-1

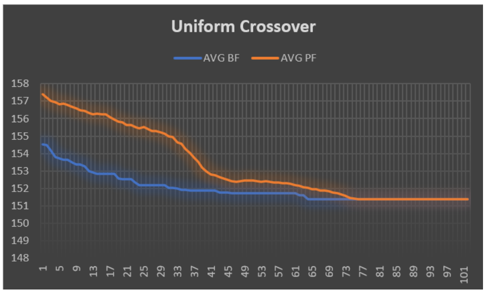

Fig. 14: Uniform Crossover Doc-1

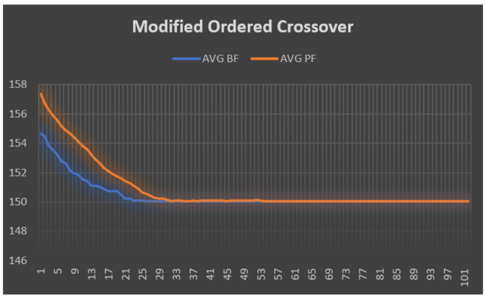

Fig. 15: Modified Ordered Crossover Doc-1

	Average Best	Population Best
Final Fitness	149.7407233	150.0442347
Min	149.7407233	150.0063329
Max	154.2647393	157.4362923
Mean	150.5327412	152.298131
Median	150.0348058	151.8066648
Standard Deviation	1.120867058	1.922387551

TABLE V: Figure 13 Table

	Average Best	Population Best
Final Fitness	150.0221384	150.0221384
Min	150.0221384	150.0221384
Max	154.6485058	157.3905809
Mean	150.4230646	150.8909237
Median	150.0221384	150.0587843
Standard Deviation	0.980592658	1.751209566

TABLE VI: Figure 15 Table

D. Experiment 4

Just like the experiments above, the modified ordered crossover resulted in convergence at earlier generation (see Fig. 18) while ordered crossover took the longest to converge (see Fig. 16) but gave the best fitness (see Table 7,8).

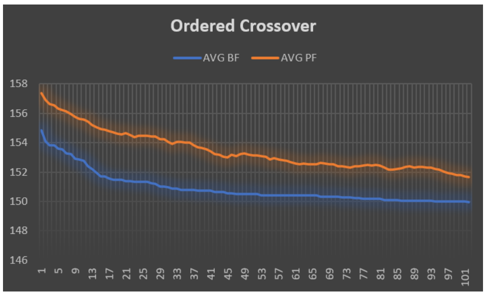

Fig. 16: Ordered Crossover Doc-1

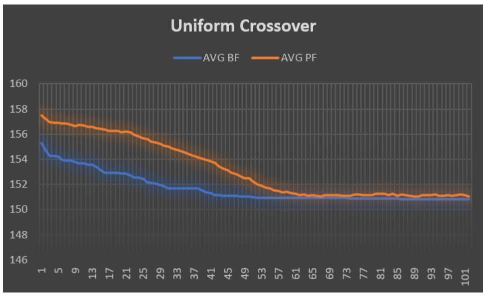

Fig. 17: Uniform Crossover Doc-1

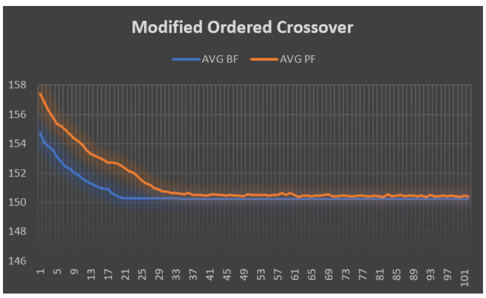

Fig. 18: Modified Ordered Crossover Doc-1

	Average Best	Population Best
Final Fitness	149.9699213	151.6505226
Min	149.9699213	151.6505226
Max	154.8560127	157.3855842
Mean	150.9057168	153.4716996
Median	150.5184904	153.0662712
Standard Deviation	1.071391898	1.350625269

TABLE VII: Figure 16 Table

	Average Best	Population Best
Final Fitness	150.2257236	150.4094977
Min	150.2257236	150.3677927
Max	154.747928	157.4170595
Mean	150.5839879	151.324115
Median	150.2257236	150.5065357
Standard Deviation	0.914715593	1.643117449

TABLE VIII: Figure 18 Table

E. Experiment 5

For document 1, the modified ordered crossover performed slightly better compared to the ordered crossover by having the overall best final fitness and convergence time (see table 9,10 and figure 19, 21), but for document 2 the results were once again the same as the previous experiments where modified ordered crossover had the best convergence (see Fig 22), ordered crossover had the best overall fitness (see Table 11), and uniformed crossover had the worst fitness but a better convergence compared to ordered (see Fig 20 and table 12). This results also holds for document 3, meaning the modified ordered crossover was ”lucky” to find a better fitness when compared to ordered crossover.

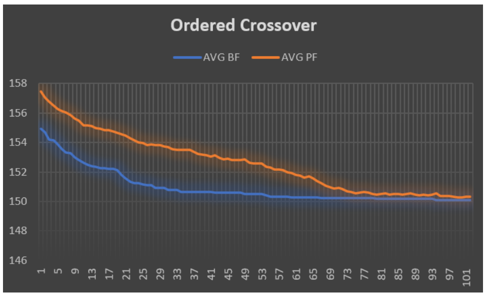

Fig. 19: Ordered Crossover Doc-1

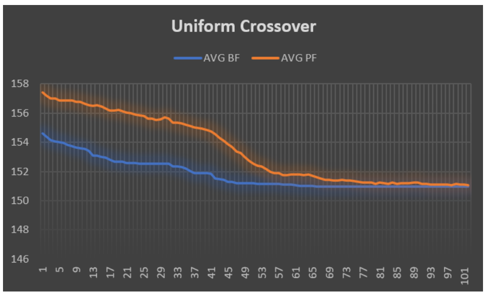

Fig. 20: Uniform Crossover Doc-1

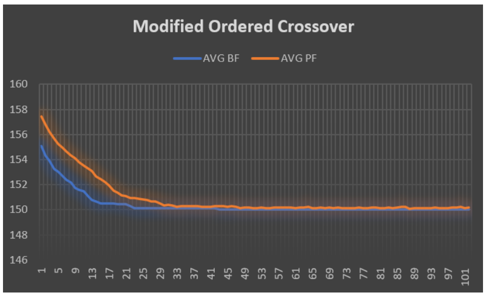

Fig. 21: Modified Ordered Crossover Doc-1

	Average Best	Population Best
Final Fitness	150.1000335	150.3014293
Min	150.1000335	150.2536028
Max	154.9353784	157.4477318
Mean	150.9480593	152.6249107
Median	150.517245	152.5915941
Standard Deviation	1.145131016	1.947068719

TABLE IX: Figure 19 Table

	Average Best	Population Best
Final Fitness	150.0253155	150.1893436
Min	150.0253155	150.0784312
Max	155.0738899	157.4497212
Mean	150.4070122	150.9521566
Median	150.0253155	150.2077065
Standard Deviation	0.953257246	1.641423328

TABLE X: Figure 21 Table

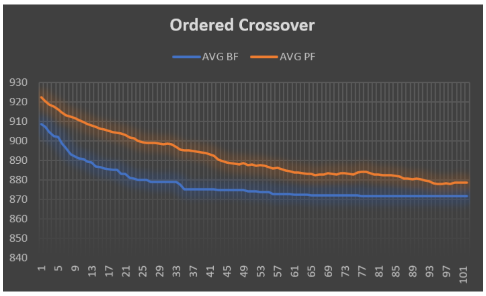

Fig. 22: Ordered Crossover Doc-2

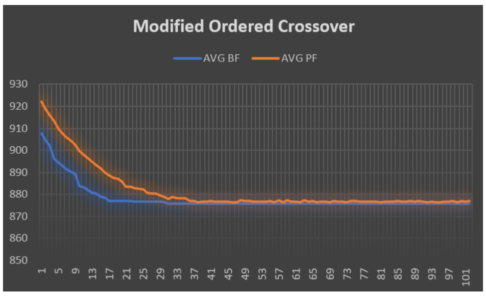

Fig. 23: Modified Ordered Crossover Doc-2

	Average Best	Population Best
Final Fitness	871.8170715	878.7366481
Min	871.8170715	877.922535
Max	908.6500607	922.6518053
Mean	877.7712769	891.8317368
Median	874.2040193	887.5164501
Standard Deviation	8.822526164	11.73970291

TABLE XI: Figure 22 Table

	Average Best	Population Best
Final Fitness	875.8163437	876.9504916
Min	875.8163437	876.2013716
Max	907.8143898	922.1379009
Mean	878.1350766	881.9427983
Median	875.8163437	876.9003656
Standard Deviation	6.232374632	10.58301112

TABLE XII: Figure 23 Table

V. Conclusions

From the five experiments performed using different parameters on ordered crossover, uniform crossover, and the modified ordered crossover, the ordered crossover always, within a margin of error, resulted in having the best final fitness. Whereas the modified ordered crossover reached convergence at a lower generation span but the final fitness was higher then the ordered crossover. Finally, the uniformed crossover reached convergence earlier then the ordered crossover but not the modified ordered crossover. It also produced the worst overall final fitness when compared to the other crossovers. Thus, it can be concluded that the ordered crossover is better if the desired result is only the best fitness and not the amount of time needed to find the best fitness. Otherwise, the better crossover would be the modified ordered crossover, if a good, not necessarily the best, fitness is needed in a short amount of time.

References

[1] V. Mallawaarachchi, “Introduction to Genetic Algorithms - Including Example Code,” Medium, 01-Mar-2020. [Online]. Available: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3. [Accessed: 30-Oct-2020].

[2] “Mutation (genetic algorithm),” Wikipedia, 16-Feb-2020. [Online]. Available: https://en.wikipedia.org/wiki/Mutation_(genetic_algorithm). [Accessed: 30-Oct-2020].

[3] Nature News. [Online]. Available: https://www.nature.com/scitable/knowledge/library/mating-systems-in-sexual-animals-83033427/. [Accessed: 13-Nov-2020].