GENETIC PROGRAMMING TEST 1º
Applying "Sequence Metric"

ELEMENTS THAT WILL CHANGE DURING TEST:

VARIABLES	TECHNIQUES
Population size	cut
Crossover number	slipcut
Mutation (Standard)	slipcutup
Mutate Number	peal
Mutate Operation	pealup
Initial Depth of Trees	infaro
Length Max of Tree	infaroup
	outfaro
	outfaroup

First we will test how well our GP model performs using the same variable inputs and just changing the techniques. This way we will be able to evaluate if the performance of our algorithm varies depending on the techniques.

SETUP:
The setup for the first test is the following:

Population size	1000
Crossover number	500
Mutation (Standard)	200
Mutate Number	0
Mutate Operation	0
Initial Depth of Trees	4
Length Max of Tree	20

So from a total of 100%, 50% will be crossovered and 20% of the crossover will be mutated.

Our **objective function** will be the **distance** between the **output deck** and the **desired deck**.
As the algorithm is stochastic each test will be **run three times** to get the **average answer**.
The metric to be used is "**Sequence Metric**" (specific distance created for this problem).

1. PART: TECHNIQUE ANALYISIS:

We have to analize the performance of the algorithm depending the techniques taking into account the next conditions: **number of techniques**, **order of techniques**, and **used techniques**.

FIRST TEST: (goal: **test** the performace of **each technique separately**)
For this experiment we will create as outputs different decks of cards (ordenations) using different techniques and we will see how well our algorithm performs. Example: we have a deck "A" and we will apply for example two cuts (cutting in position 13 and later in position 27 as an example), thus, we will get a deck "B". Now we can see if or algorithm finds the proper way to reach from deck "A" to deck "B" which should be two cuts (with the numbers 13 and 27).

We will change the **number of techniques** but we will NOT take into account the **order of techniques** or **used techniques**, thus, we will just use one technique at a time.

Experiment Model:
We will just use one technique at a time and we will increase the amount of that same technique, (the numbers on left of the techniques mean the number of times that we will apply that technique to the deck: 2, 4, 8, 16, 32 times to get the corresponding output, and then we will run our algorithm to see if with the input deck "A" it finds the output deck "B").

We will analyze our output depending on two variables. **Punctuation** and **number of generation** we achieved each punctuation:

Experiment 1: cutting

cut	2	cut	4	cut	8	cut	16	cut	32

Expected output: we expect the performance will always remain equal because no matter how many times you cut the deck, the deck order is never altered, just displaced from its original position.

Experiment 2: slipcutting

slipcut	2	slipcut	4	slipcut	8	slipcut	16	slipcut	32

Expected output: taking into account that our model analizes sequences, a slipcut is just displacing one card to a random place (like an insert on an array), thus, our model may possibly not work very well. We also expect that while increasing number of times we apply a technique, the model will have more difficulty finding a proper solution.

Experiment 3: slipcutup-ing

slipcutup	2	slipcutup	4	slipcutup	8	slipcutup	16	slipcutup	32

Expected output: we expect same output as slipcut because its the same principle but with the deck facing up.

Experiment 4: pealing

peal	2	peal	4	peal	8	peal	16	peal	32

Expected output: our model is keen on finding sequences so it is supposed to work better than slipcutting, however, if we increase the amount of times we apply pealings the model may have more difficulty finding a proper solution.

Experiment 5: pealup-ing

pealup	2	pealup	4	pealup	8	pealup	16	pealup	32

Expected output: we expect same output as pealing because its the same principle but with the deck facing up.

Experiment 6: infaro-ing

infaro	2	infaro	4	infaro	8	infaro	16	infaro	32

Expected output: again, our model is keen on finding sequences so it is supposed to work better than slipcutting and in a similar way to pealing, also, if we increase the amount of times we apply infaros the model may have more difficulty finding a proper solution.

Experiment 7: infaroup-ing

infaroup	2	infaroup	4	infaroup	8	infaroup	16	infaroup	32

Expected output: we expect same output as infaro-ing because its the same principle but with the deck facing up.

Experiment 8: outfaro-ing

outfaro	2	outfaro	4	outfaro	8	outfaro	16	outfaro	32

Expected output: we expect same output as infaro-ing because its the same principle but with the first packet above the left packet.

Experiment 9: outfaroup-ing

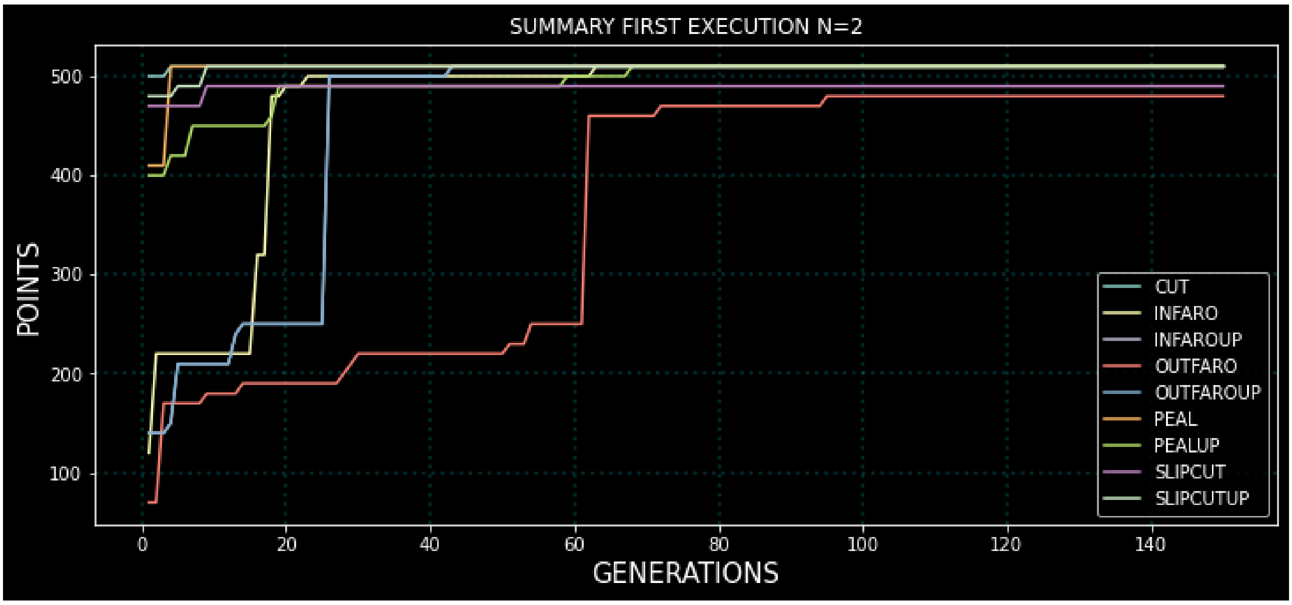
outfarou p	2	outfarou p	4	outfarou p	8	outfarou p	16	outfarou p	32

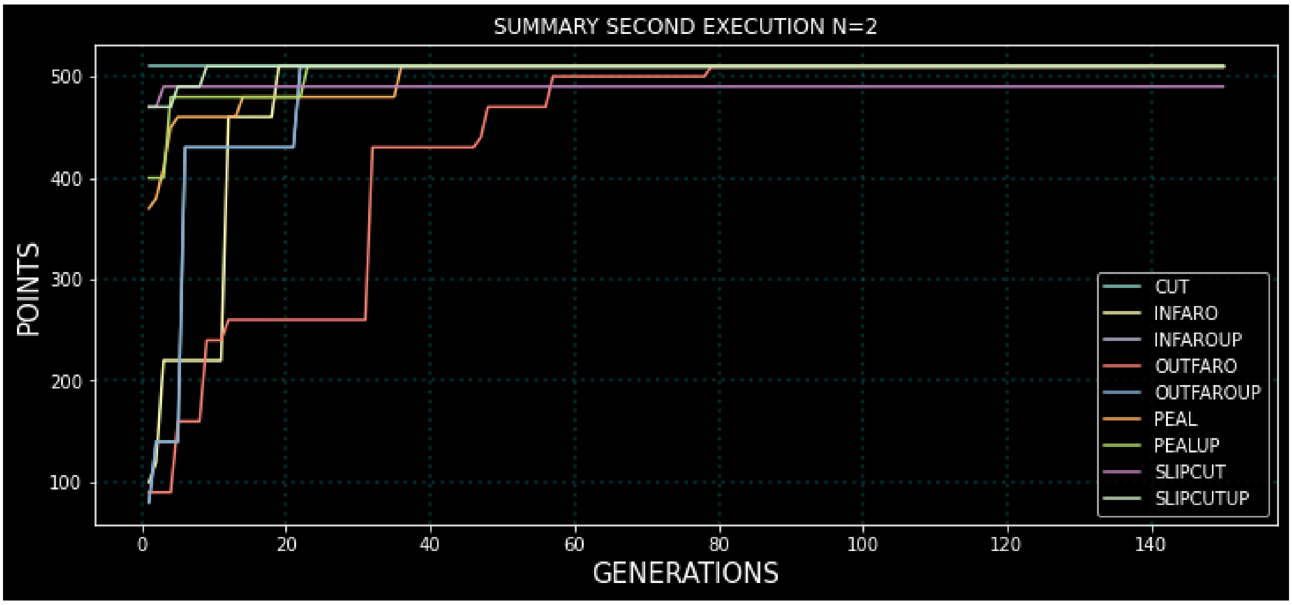
Expected output: we expect same output as outfaro-ing because its the same principle but with the deck facing up.

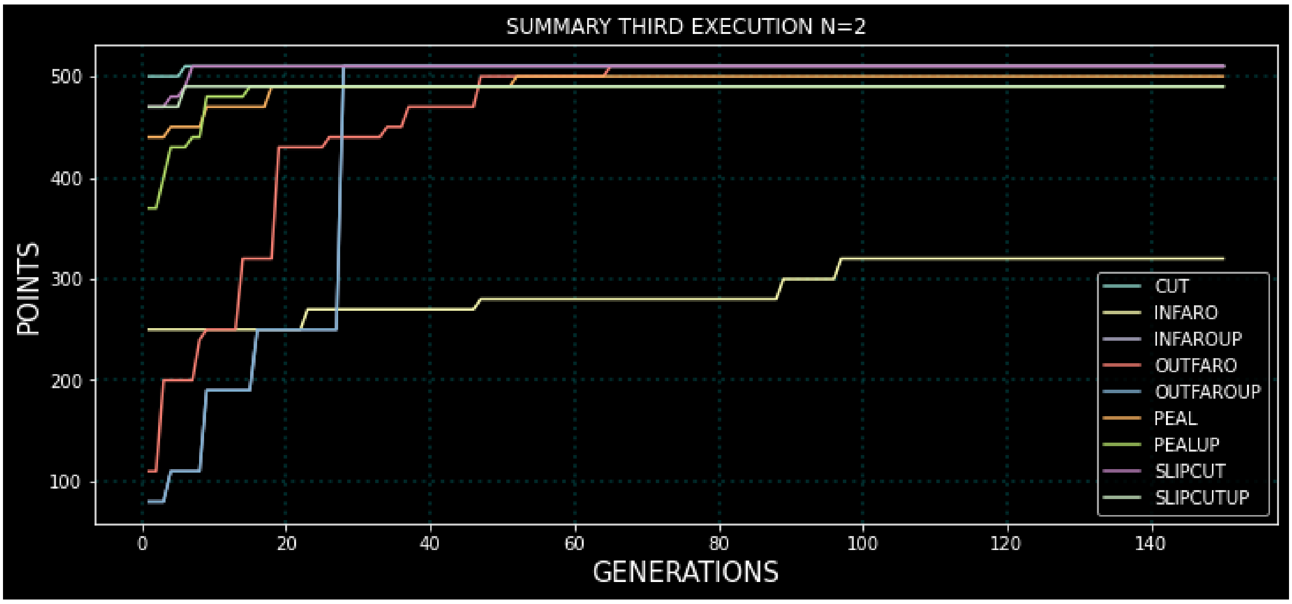
Hypothesis of outputs: on **insertion** (infaro, outfaro, infaroup and outfaroup)and **inversion** (peal and pealup) techniques we expect a **good performance** as our model is proper for analyzing sequences and each time we **increase number of techniques** will probably get **harder**., on **cutting** we expect **always** the same **good performance** as explained before, on **insertion** (slipcut and slipcutup) techniques we expect probably a **bad performance** and while increasing how many times we apply a technique we expect **performance** to **get worse**.

FIRST TEST CONLUSIONS:

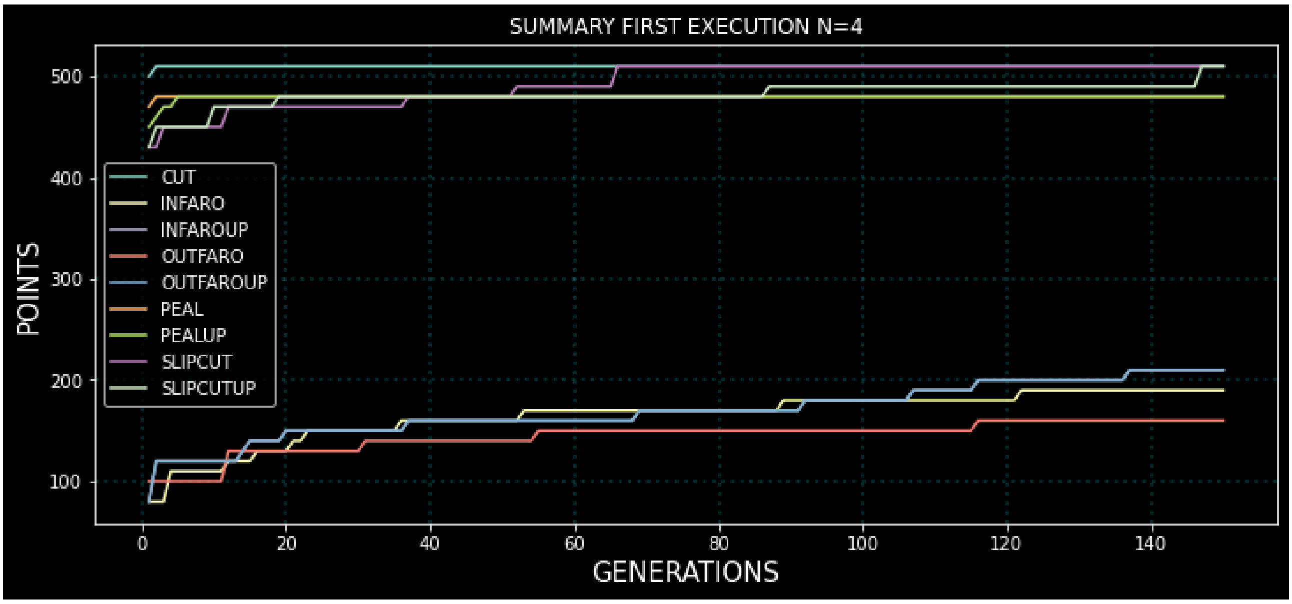
N=2 Executions:

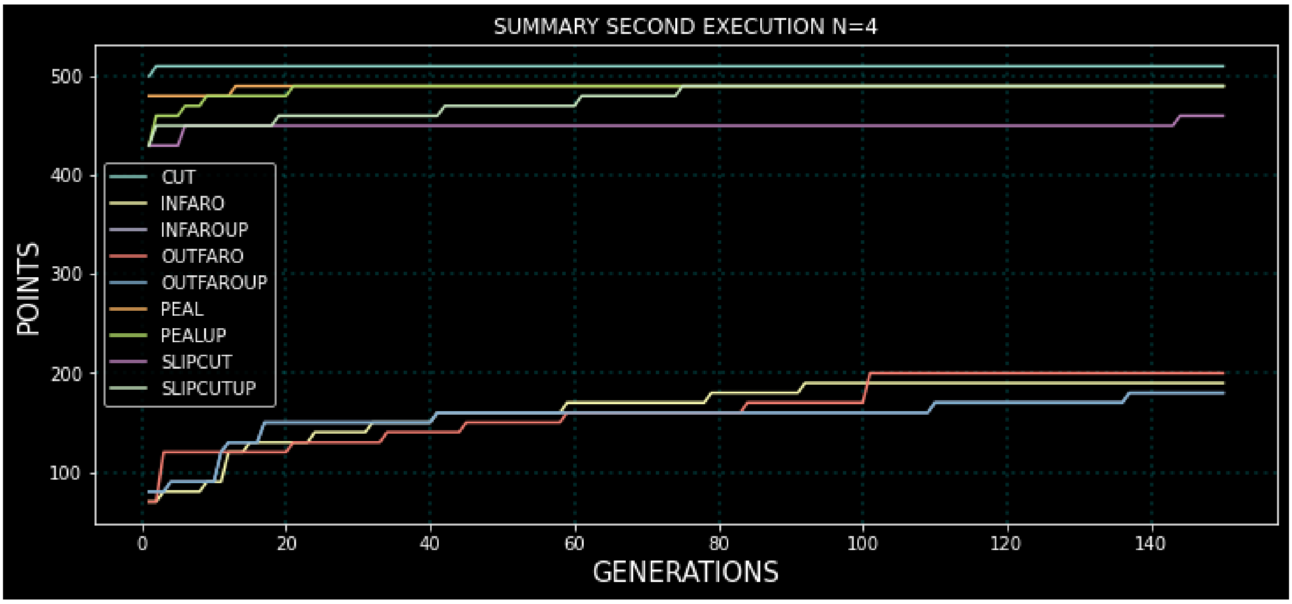


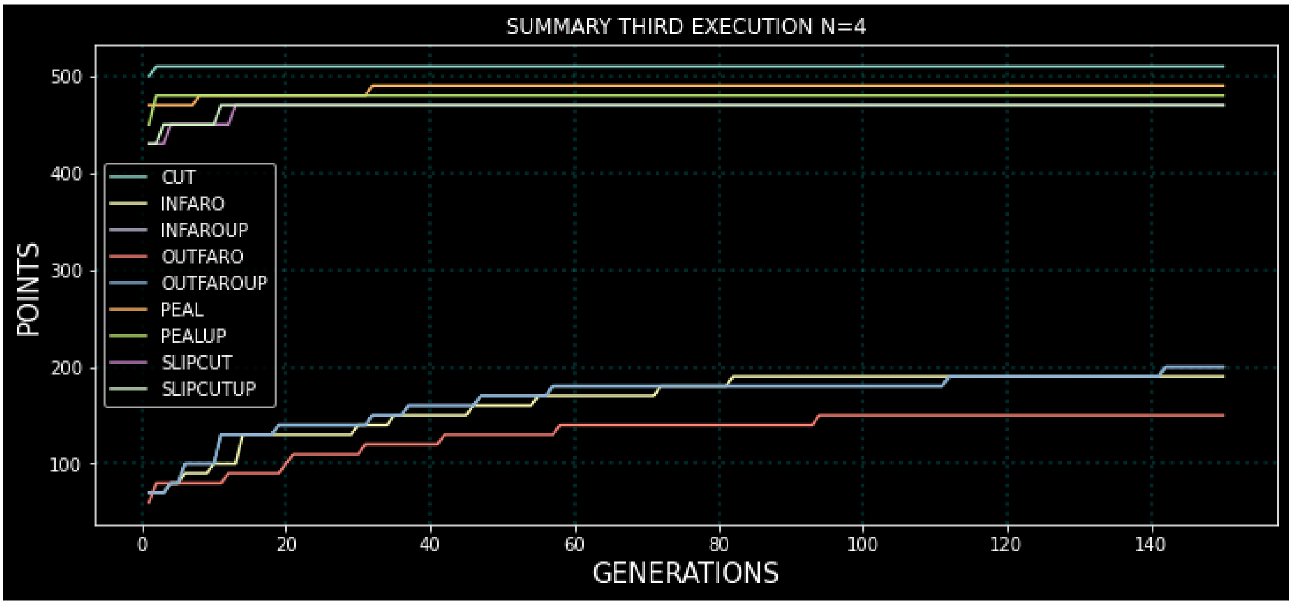




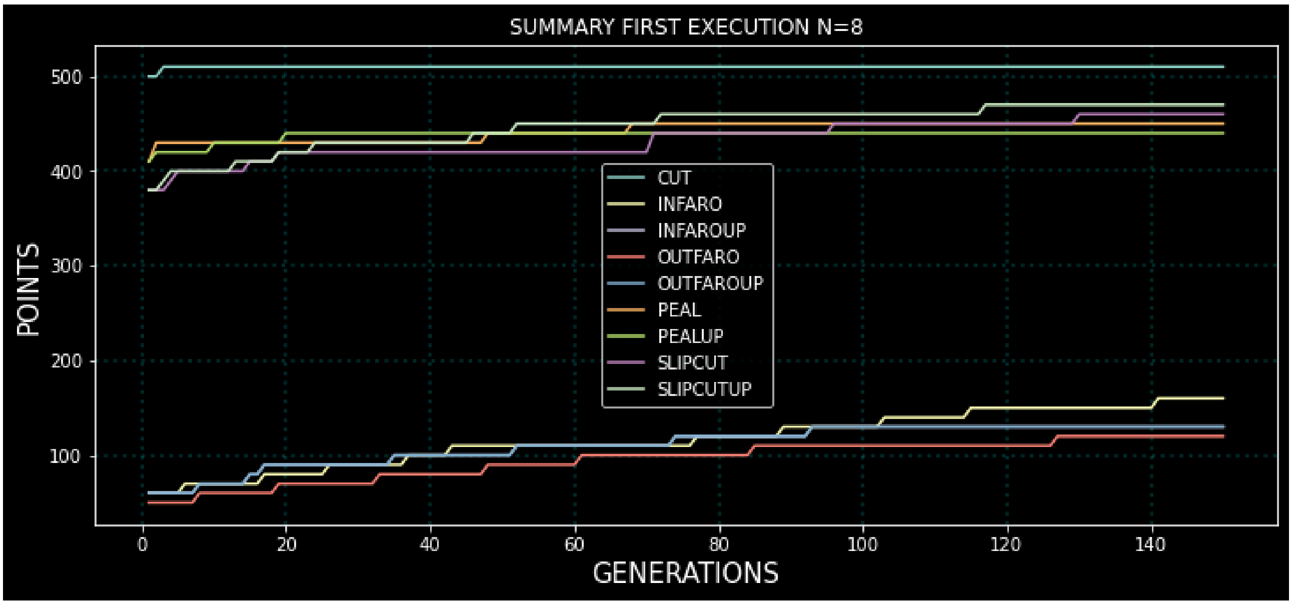
N=4 Executions:

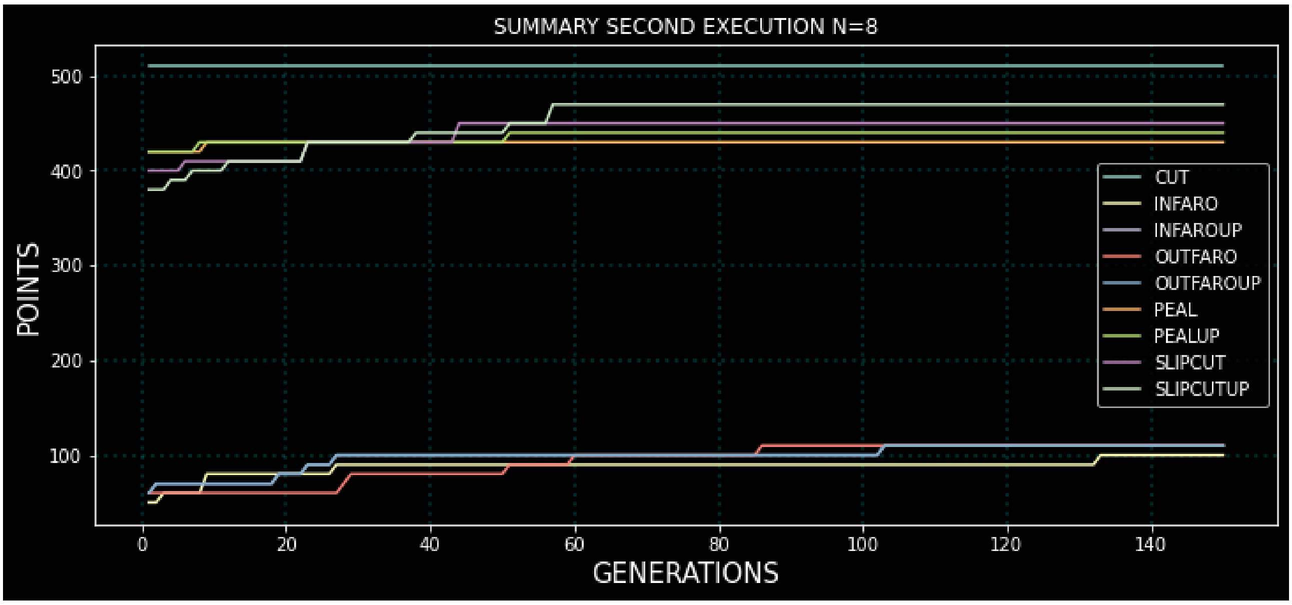


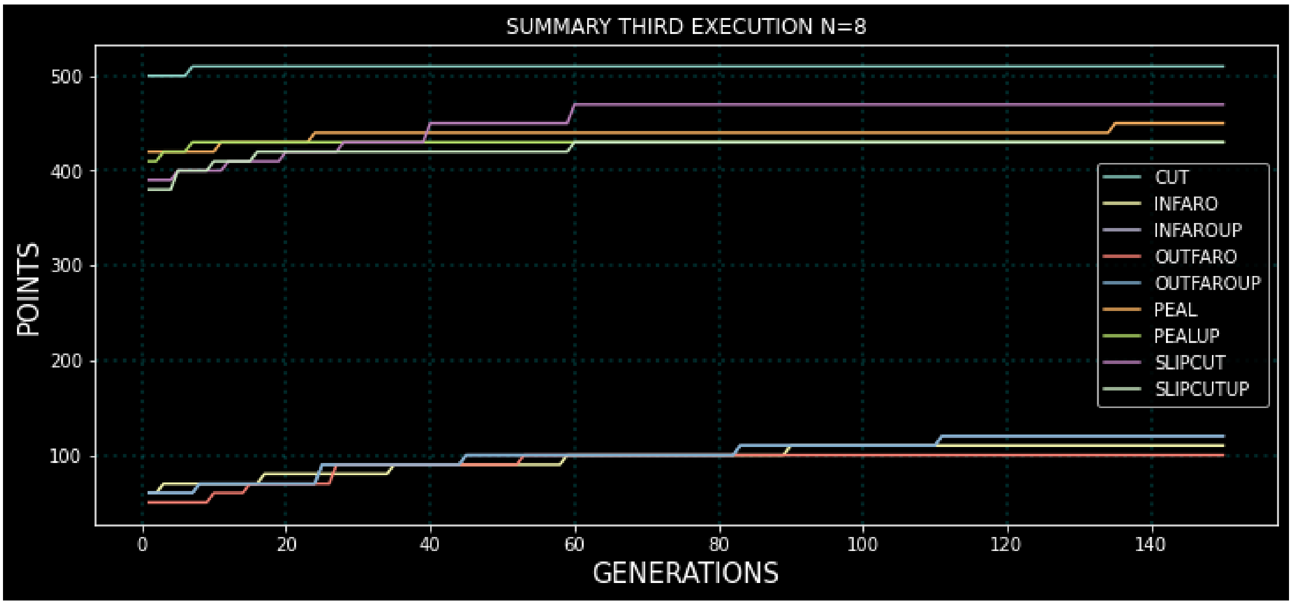




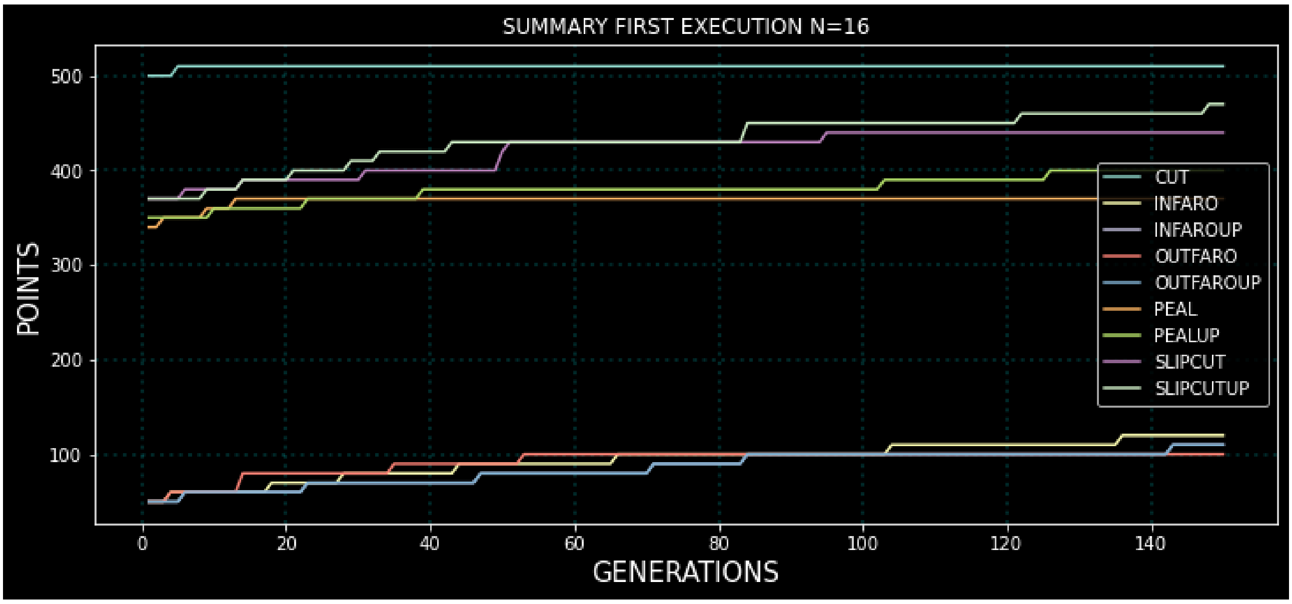
N=8 Executions:

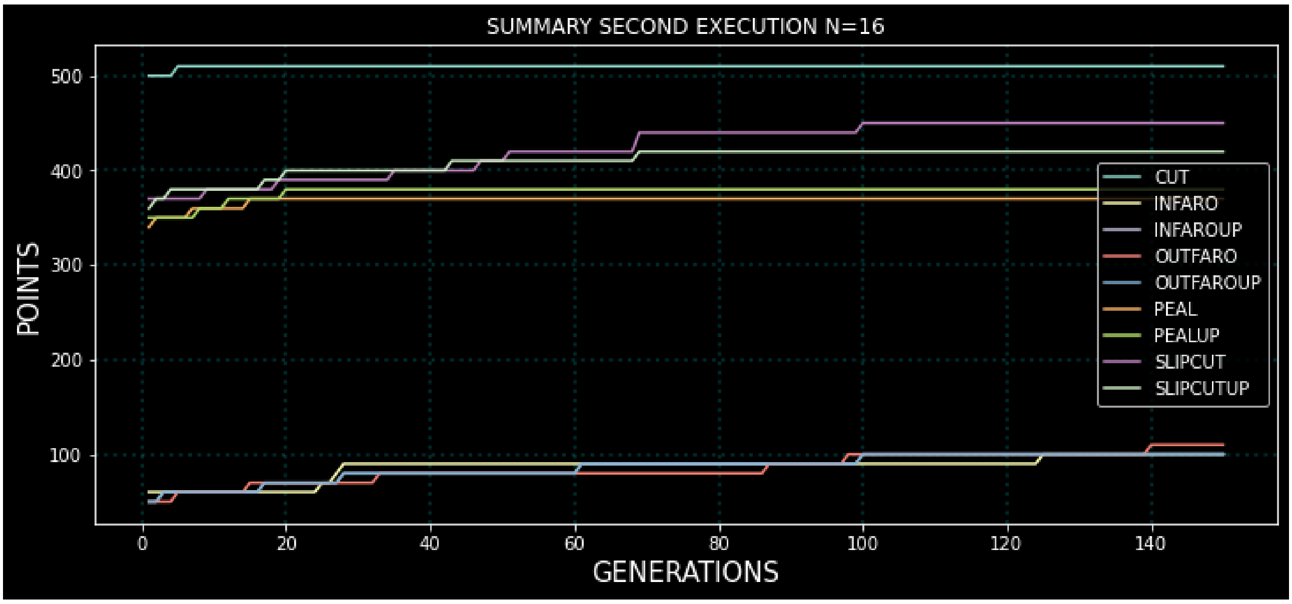


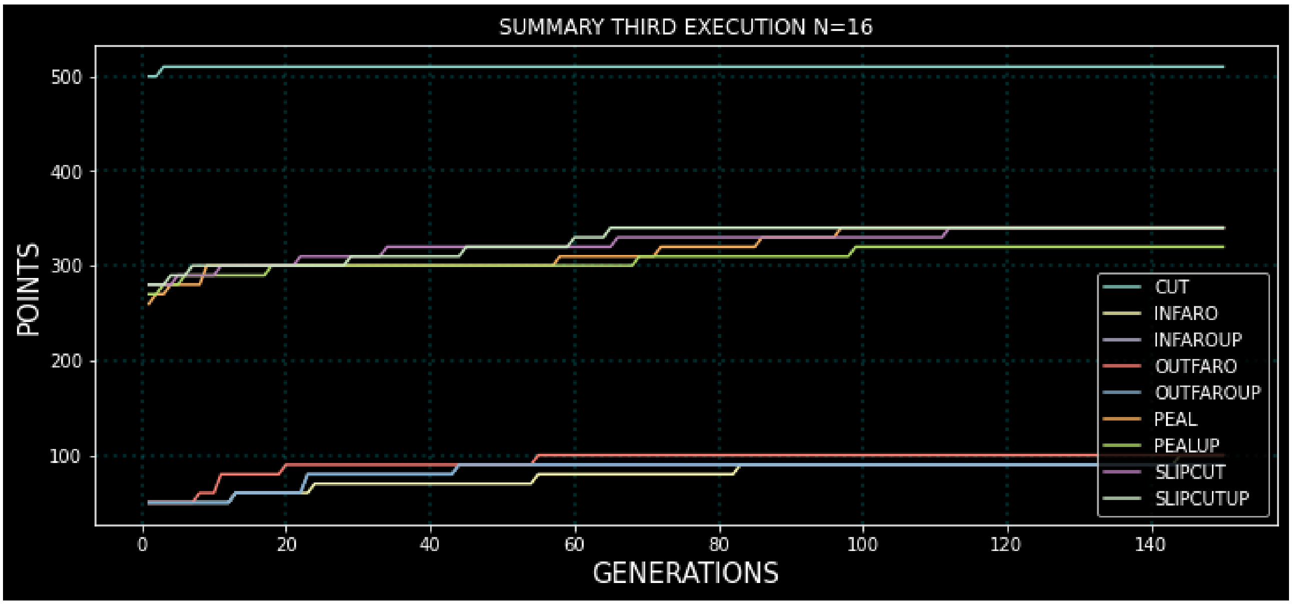




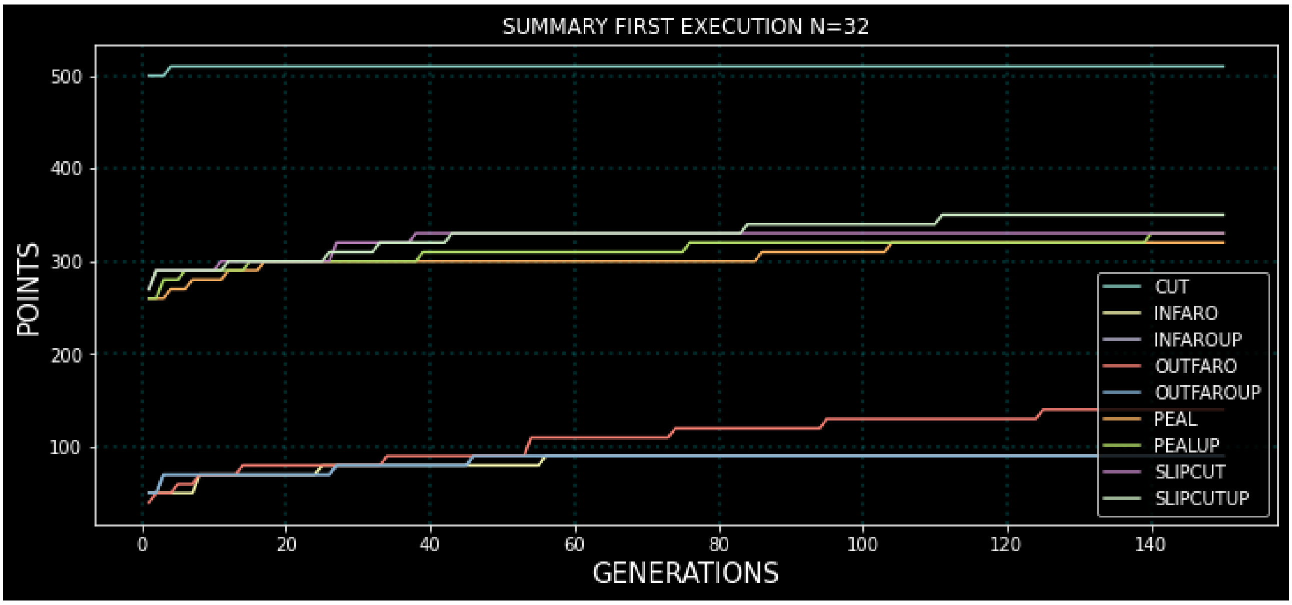
N=16 Executions:

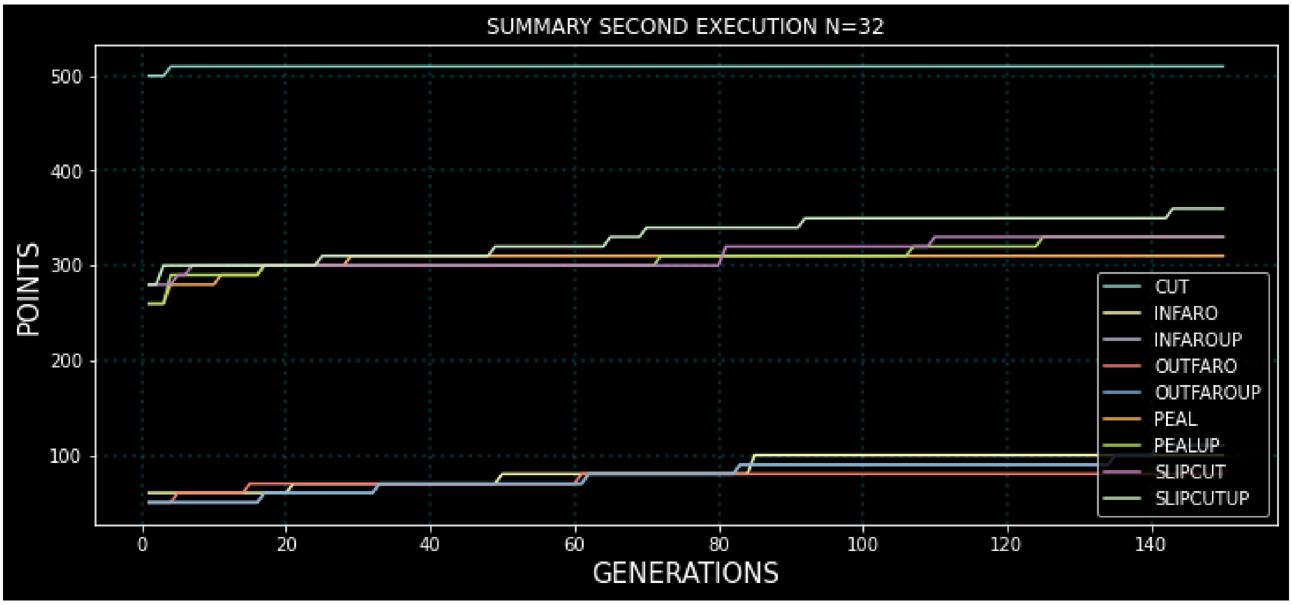


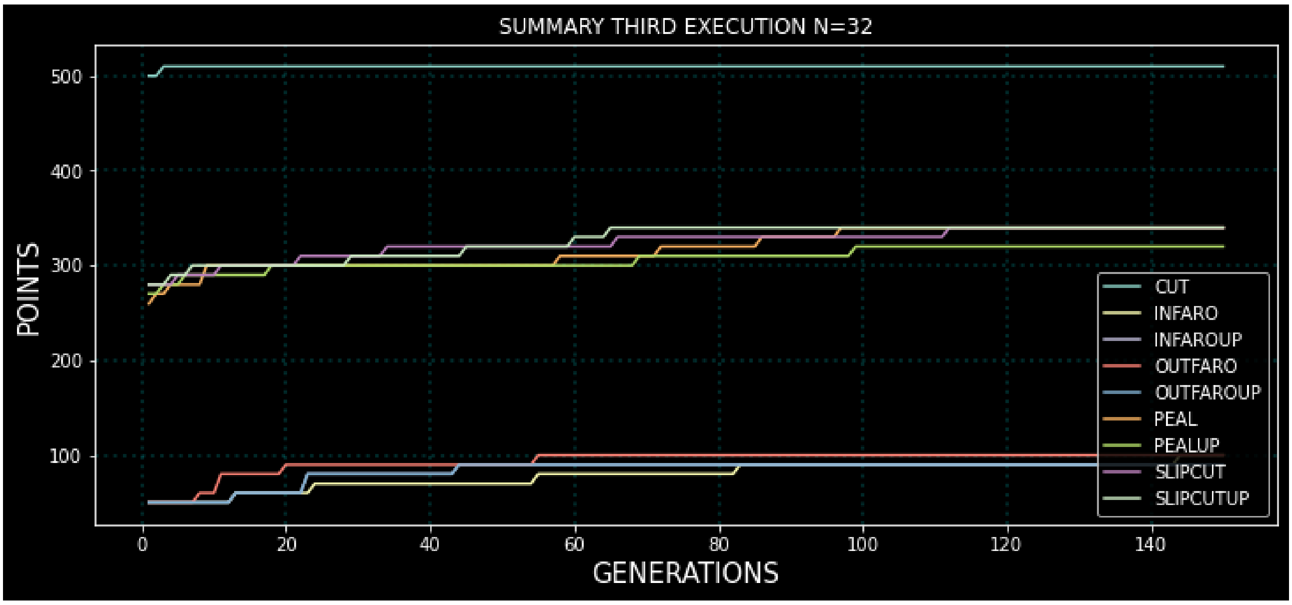




N=32 Executions:







Now we will arrive to real conclusions based in our previous hypothesis:

<u>First Conclusion</u>:
Hypothesis:
First of all, we separated the techniques in four groups because the hypothesis was that each group of techniques will perform in a similar way: **cutting**, **intercalation**, **inversion** and **insertion**.

Conclusion:
Every technique inside a group of techniques will perform in a similar way. **Intercalation** (infaro, outfaro, infaroup and outfaroup perform in a very similar way), **Inversion** (peal and pealup perform in a very similar way), **insertion** (slipcut and slipcutup perform in a similar way) and finally **cutting** its a technique in its own.

This can be used to accelerate the experiment, now we do not necessarily need to analyze nine techniques, we can just analyze 4 technique types to extract conclusions.

Lets analyze now each technique type to find new conclusions:

<u>Cutting Conclusion:</u>
Hypothesis:
We expect the performance will always remain equal because no matter how many times you cut the deck, the deck order is never altered, just displaced from its original position.

Conclusion:
The hypothesis is perfect, if we analyze all the graphics no matter N=2,4,8,16,32 in the first generations the optimal solution is found. So in case of cutting the results are brilliant.

<u>Insertion Conclusion:</u>
Hypothesis:
Taking into account that our model analizes sequences, a slipcut is just displacing one card to a random place (like an insert on an array), thus, our model may possibly not work very well. We also expect that while increasing number of times we apply a technique, the model will have more difficulty finding a proper solution.

Conclusion:
Even if our model analyzes sequences it does pretty well with insertion and as expected as the number of insertions is increased the solution is harder to find. For N=2 The solution is always found, but as the N number is increased the points go decreasing: N=4=> (Generation 150) Average Points = 500, N=8=> (Generation 150) Average Points=470, N=16=> (Generation 150) Average Points = 430, N=32 => (Generation 150) Average Points=350.

We can clearly see the decreasing of the performance

Inversion Conclusion:
Hypothesis:
Our model is keen on finding sequences so it is supposed to work better than slipcutting, however, if we increase the amount of times we apply pealings the model may have more difficulty finding a proper solution.

Conclusion: We expected the inversion to have better performance than slipcutting but it has almost the same performance (a tiny little worse, almost imperceptible): For N=2 The solution is always found, but as the N number is increased the points go decreasing: N=4=> (Generation 150) Average Points = 480, N=8=> (Generation 150) Average Points=450, N=16=> (Generation 150) Average Points = 420, N=32 => (Generation 150) Average Points=330.

Intercalation Conclusion:
Hypothesis:
Again, our model is keen on finding sequences so it is supposed to work better than slipcutting and in a similar way to pealing, also, if we increase the amount of times we apply infaros the model may have more difficulty finding a proper solution.

Conclusion: The hypothesis is completely wrong. Our algorithm is keen on finding sequences, but it looks like each time a faro shuffle is done the deck gets even more shuffled, thus, there is no pattern to find, so it has clearly the worst performace of all and as more intercalation techniques are applied the performance gets worse as expected. For N=2 The solution is always found, but as the N number is increased the points go decreasing: N=4=> (Generation 150) Average Points = 200, N=8=> (Generation 150) Average Points=120, N=16=> (Generation 150) Average Points = 100, N=32 => (Generation 150) Average Points=100.

General Conclusions of Test 1:
First, we can simplify techniques in technique groups as we have proved that they have a similar performance.

Second, we have seen that cutting is not altered through number of cuts applied, thus, its performance is brilliant.

Third, we have also seen that insertions and inversions are the second that give best performance with insertions being a bit better than inversions but almost in an imperceptible way.

Fourth, we have seen that intercalation is a complete disaster using “Sequence Metric”.

SECOND TEST: (goal: test performance of combined techniques separately (**used techniques**))
The goal of the second test is to analyze the performance mixing different techniques at pairs. As we have 9 techniques (infaro, infaroup, outfaro, outfaroup, slipcut, slipcutup, peal, pealup, cut) that would mean testing 9^2 combinations (technically speaking (9^2)-9 as reapeated combinations have already been tested) which means testing around 81 combinations. Instead we are going to make group of similar techniques as the performace may be the same:

INTERCALATION TECHNIQUES:
-infaro
-infaroup
-outfaro
-outfaroup

INSERTION TECHNIQUES:
-slipcut
-slipcutup

INVERSION TECHNIQUES:
-peal
-pealup

CUTTING TECHNIQUES:
-cut

Now instead of having 9 techniques we have grouped them into 4 groups => (4^2)-4=12 combinations:

INTERCALATION	INSERTION
INTERCALATION	INVERSION
INTERCALATION	CUTTING
INSERTION	INTERCALATION
INSERTION	INVERSION
INSERTION	CUTTING
INVERSION	INTERCALATION
INVERSION	INSERTION
INVERSION	CUTTING
CUTTING	INTERCALATION
CUTTING	INSERTION
CUTTING	INVERSION

Note: we have assumed that techniques from the same group will have the same performance thanks to the first test.

Note: for intercalation we will use: infaro; for inversion we will use peal; for insertion we will use: slipcut; for cutting we will use: cut;

Experiment Model:

Based on our conclusions extracted from the previous experiment we will now assume the following:

-our model is perfect with cutting as it only displaces the deck no matter nomber of techniques.
-our model is good with pealing but will get harder with number of techniques.
-our model is good with insertion and will also get harder with number of techniques.
-our model is bad with intercalation but will get harder with number of techniques.

Experiment 1: INTERCALATION - INSERTION

INTER	2	INTER	4	INTER	8	INTER	16	INTER	32
INSER		INSER		INSER		INSER		INSER	

Experiment 2: INTERCALATION - INVERSION

INTER	2	INTER	4	INTER	8	INTER	16	INTER	32
INVER		INVER		INVER		INVER		INVER	

Experiment 3: INTERCALATION - CUTTING

INTER	2	INTER	4	INTER	8	INTER	16	INTER	32
CUT		CUT		CUT		CUT		CUT	

Experiment 4: INSERTION - INTERCALATION

INSER	2	INSER	4	INSER	8	INSER	16	INSER	32
INTER		INTER		INTER		INTER		INTER	

Experiment 5: INSERTION - INVERSION

INSER	2	INSER	4	INSER	8	INSER	16	INSER	32
INVER		INVER		INVER		INVER		INVER	

Experiment 6: INSERTION - CUTTING

INSER	2	INSER	4	INSER	8	INSER	16	INSER	32
CUT		CUT		CUT		CUT		CUT	

Experiment 7: INVERSION - INTERCALATION

INVER	2	INVER	4	INVER	8	INVER	16	INVER	32
INTER		INTER		INTER		INTER		INTER	

Experiment 8: INVERSION - INSERTION

INVER	2	INVER	4	INVER	8	INVER	16	INVER	32
INSER		INSER		INSER		INSER		INSER	

Experiment 9: INVERSION - CUTTING

INVER	2	INVER	4	INVER	8	INVER	16	INVER	32
CUT		CUT		CUT		CUT		CUT	

Experiment 10: CUTTING - INTERCALATION

CUT	2	CUT	4	CUT	8	CUT	16	CUT	32
INTER		INTER		INTER		INTER		INTER	

Experiment 11: CUTTING - INSERTION

CUT	2	CUT	4	CUT	8	CUT	16	CUT	32
INSER		INSER		INSER		INSER		INSER	

Experiment 12: CUTTING - INVERSION

CUT	2	CUT	4	CUT	8	CUT	16	CUT	32
INVER		INVER		INVER		INVER		INVER	

Hypothesis of outputs: inversion and insertion have similar performance, so, if we combine both we should get that inverting N times or inserting N times will have the same effect as combining both techniques N times. (example: 32 inversions or 32 intercalations should get same performance as 32 inversions and intercalations (16 inversions + 16 intercalations)).

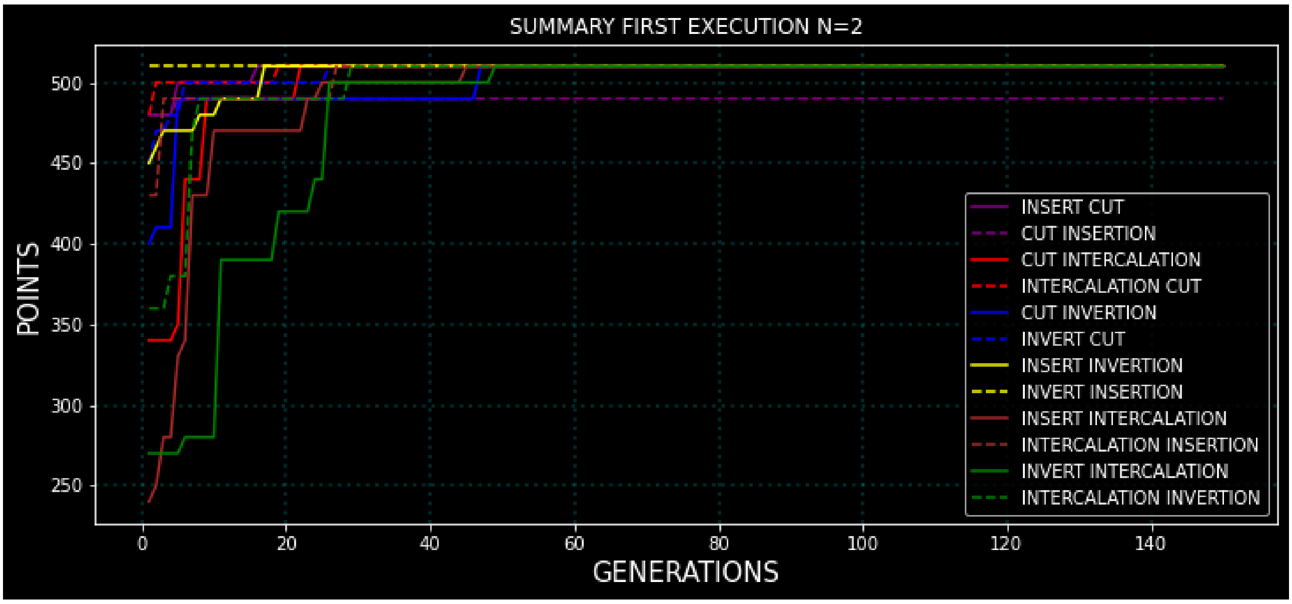
So supposedly inversions and insertions will be in the same performace type.

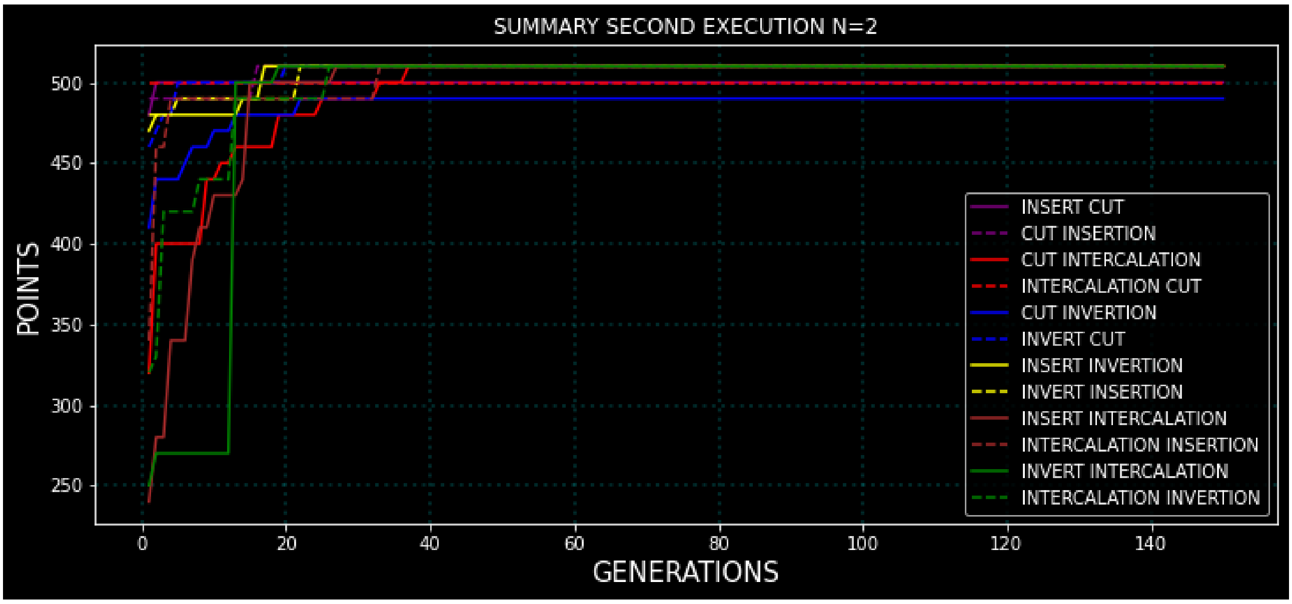
Then we have intercalation, we expect its performace to be bad and later on even worse, so, if we mix it with inversion or insertion (which performance may be good and worsen with time) we will have something like between the bad performance of intercalation and the good performance of inversions and insertions.

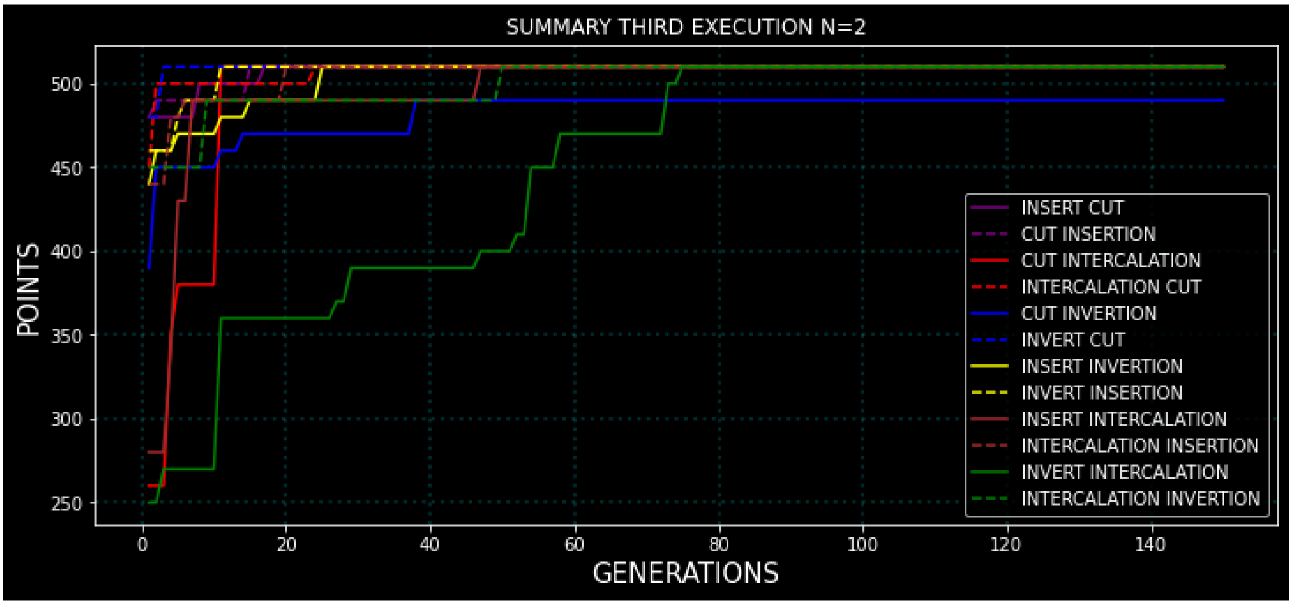
So supposedly mixing inversions or insertions with intercalations will be better than just intercalation but worse than intesertions or inversions.

Finally, we have the cutting, we dont know how the cutting will affect the other two groups (insertions – inversions and intercalations).

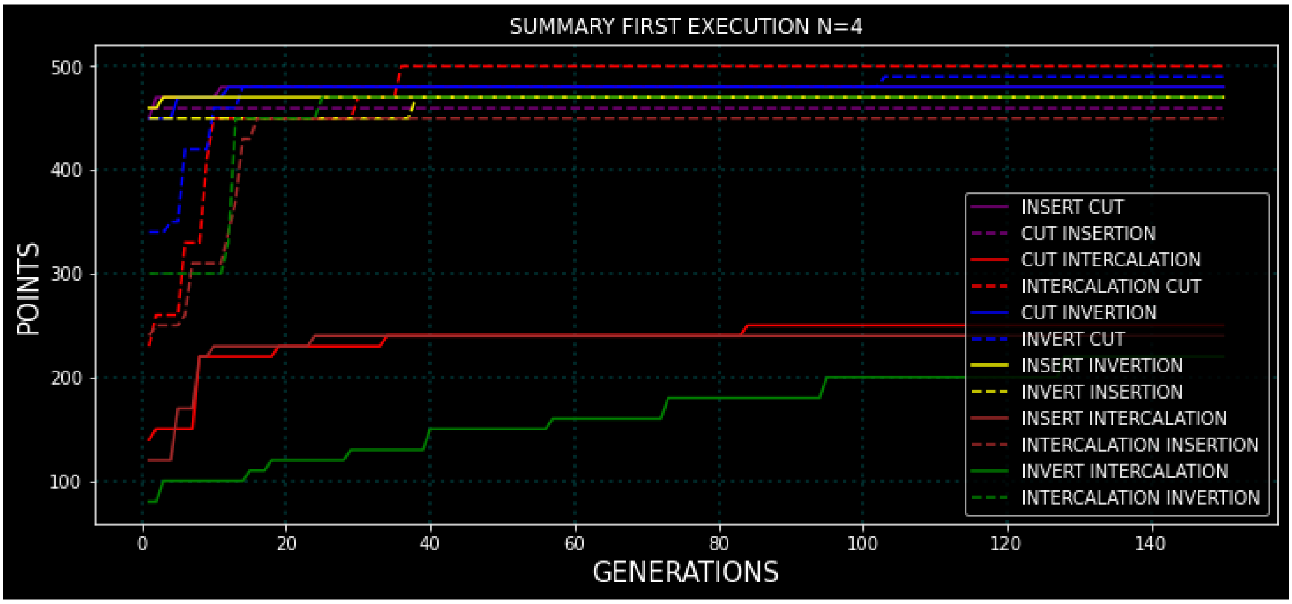
SECOND TEST CONLUSIONS:
N=2 Executions:

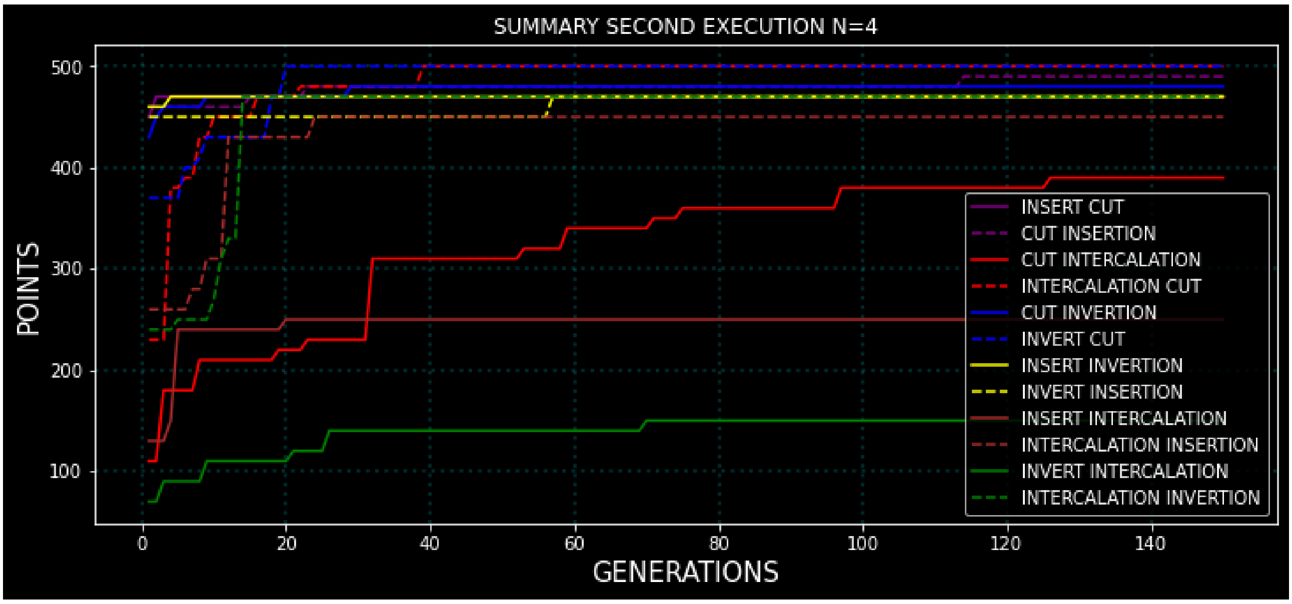


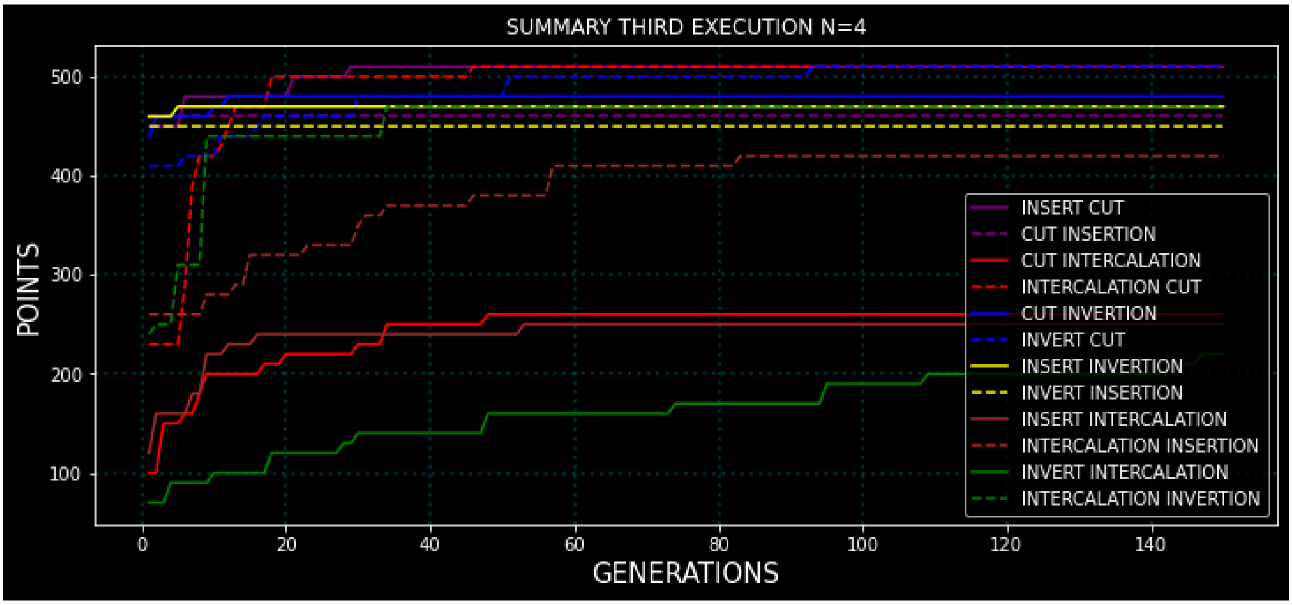




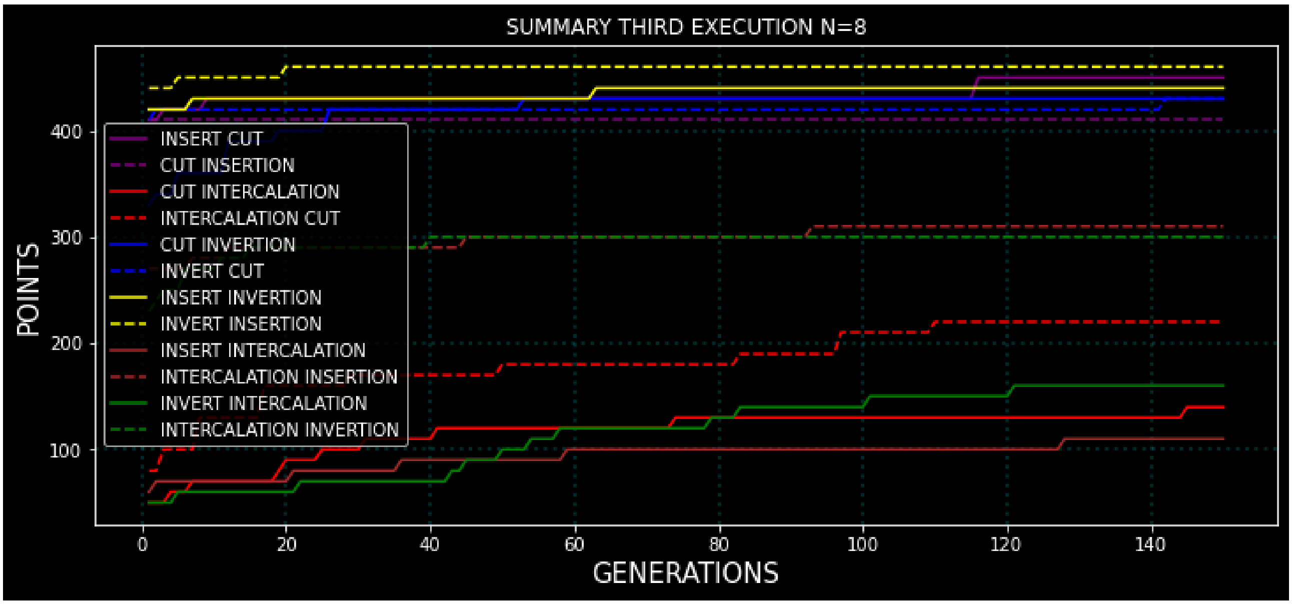
N=4 Executions:



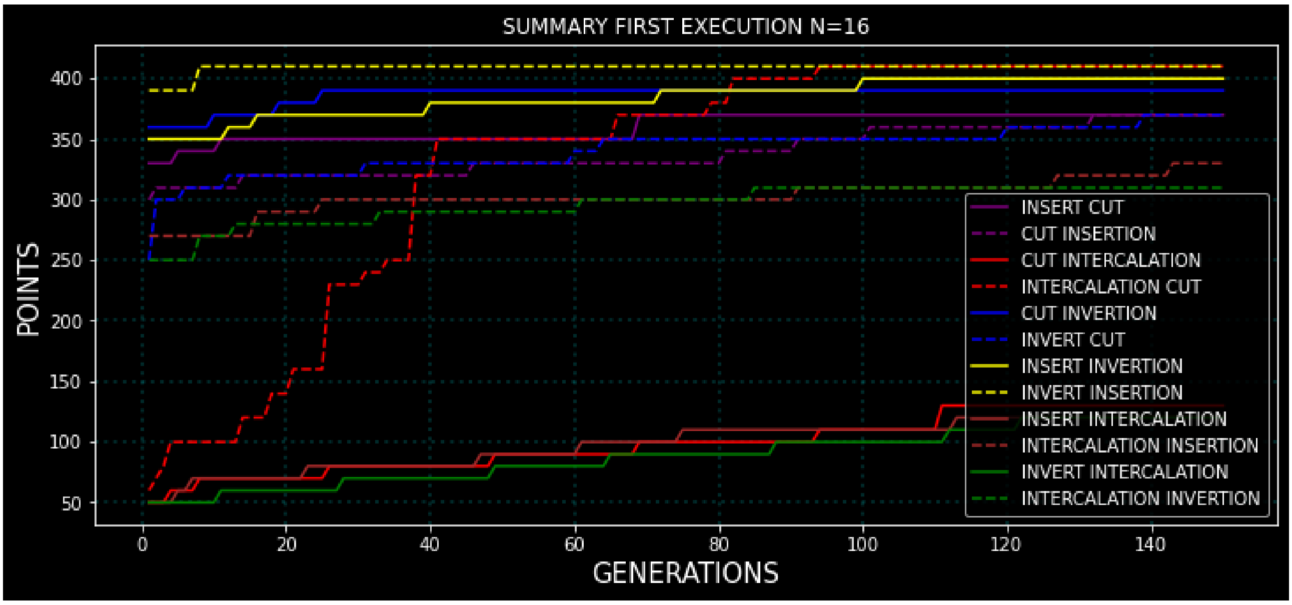


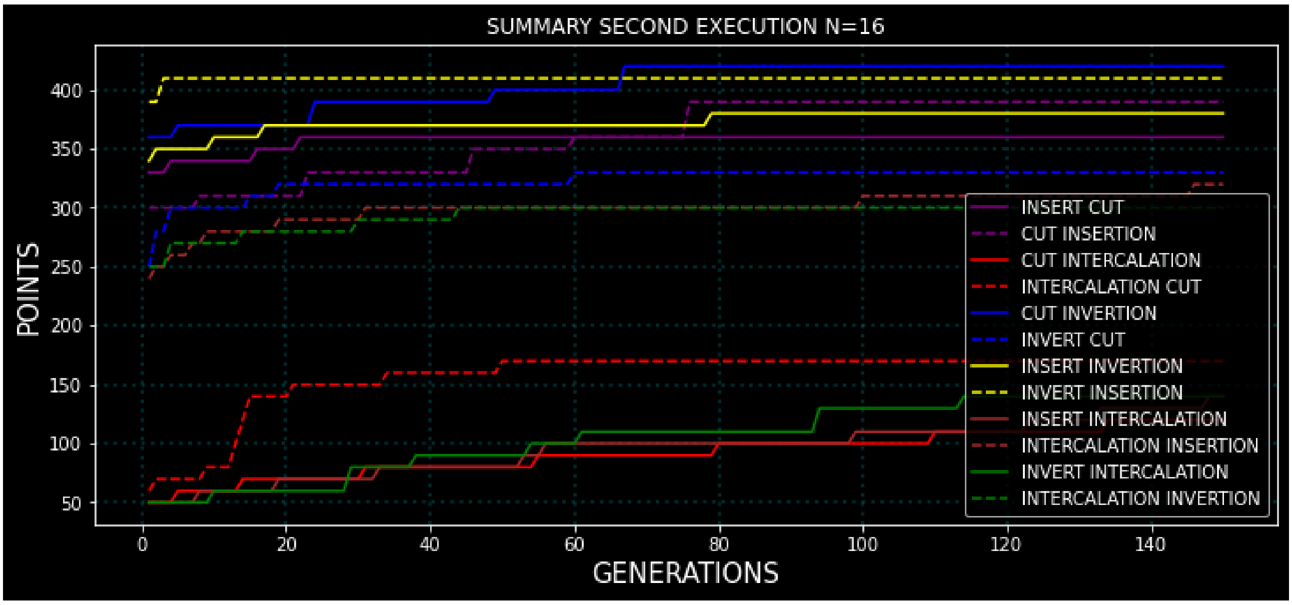


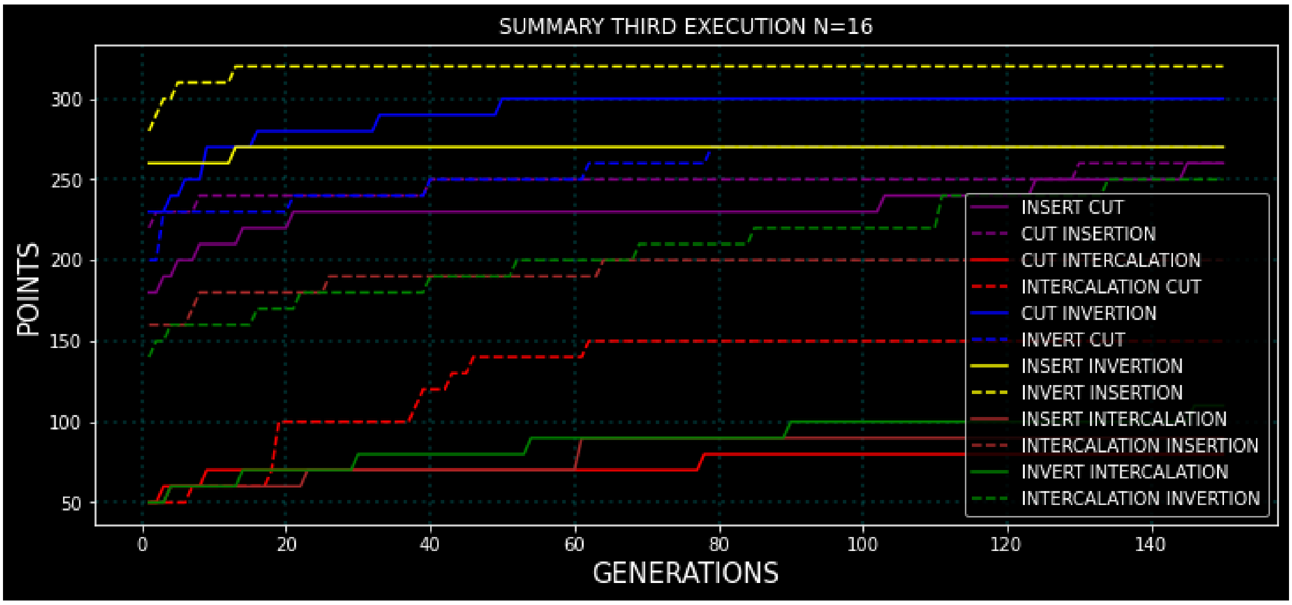
N=8 Executions:



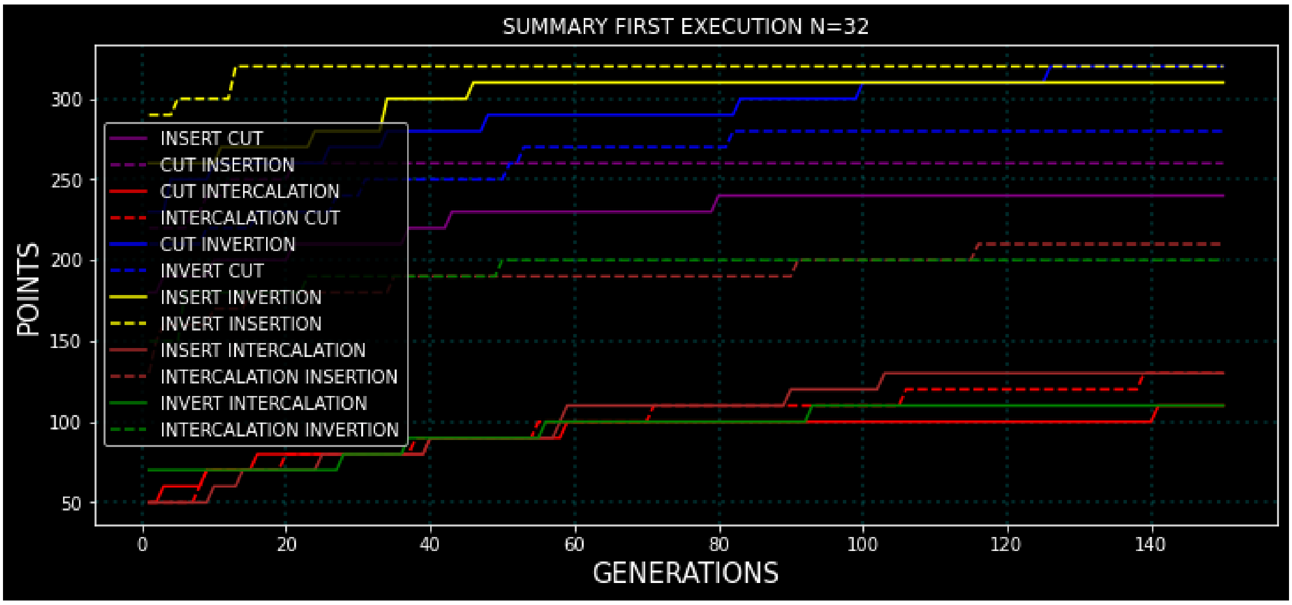
N=16 Executions:

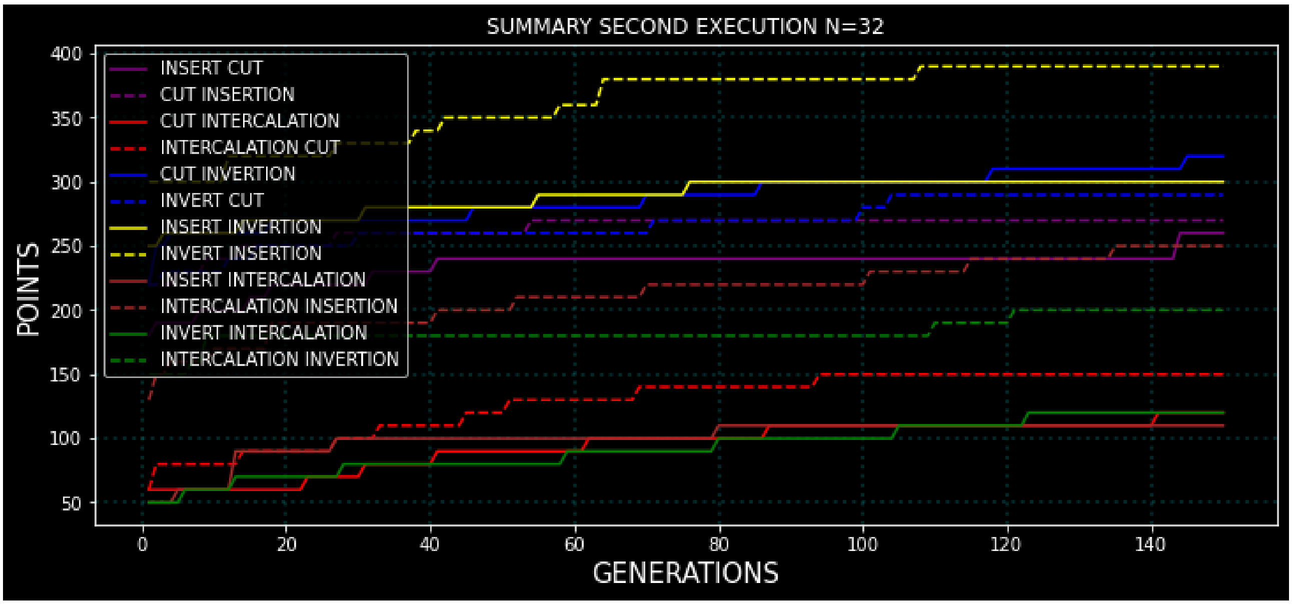


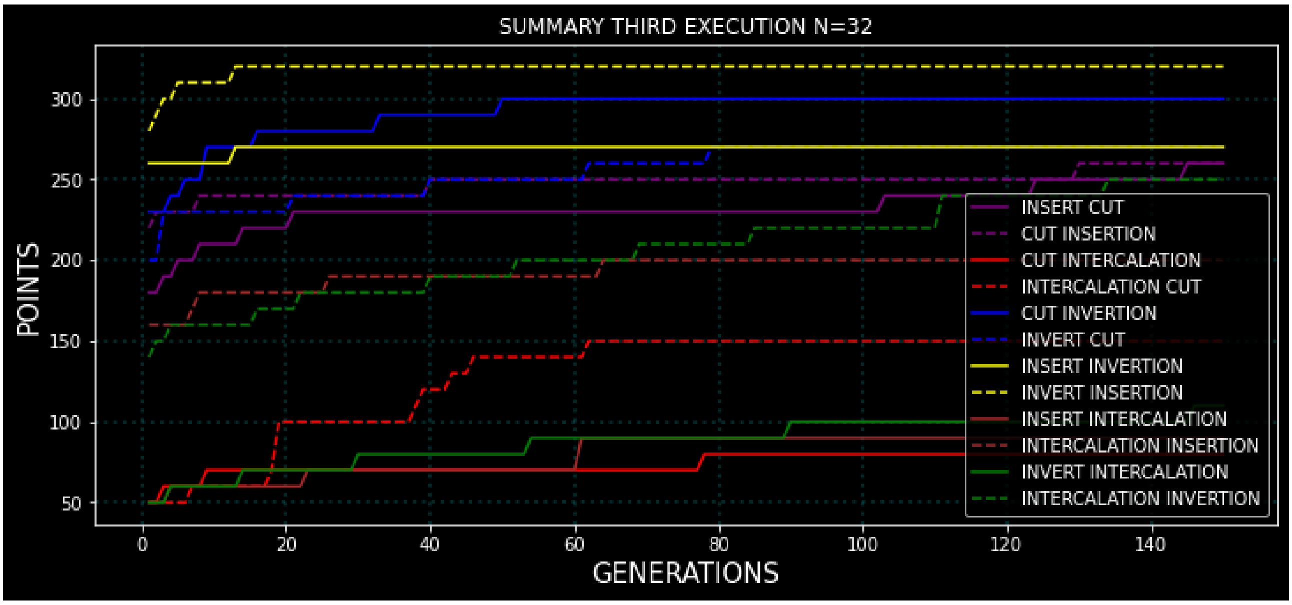




N=32 Executions:







NOTE: We have three groups of performance: cutting / inversions and insertions / intercalations. Ordered from best to worst performance as we concluded earlier in test 1.

First Conclusion:
Hypothesis:
So inversions and insertions are in the same performace type, thus, if we combine them both we should get the same performace as having them separated.
example: 32 inversions or 32 intercalations should get same performance as 32 inversions and intercalations (16 inversions + 16 intercalations)

Conclusion:
It is correct, the performance in just insertions, or just inversions or insertions+inversions in both cases is around 320 points (generation 150 | N=32) . In conclusion {points(32 insertions)=points(32 inversions)=points(16 insertions+16 inversions) | points is the punctuation at generation 150}

Second Conclusion:
Hypothesis:
Then we have intercalation, we expect its performace to be bad and later on even worse, so, if we mix it with inversion or insertion (which have the same performace as stated before in test 1) we will have something like between the bad performance of intercalation and the good performance of inversions and insertions.

Conclusion:
It is again correct, if we take for example N=32 (Generation 150) we can see that intercalation scores around 100 points whereas insertion or inversion 300 points. If we see the points of intercalation-insertion, insertion-intercalation, intercalation-inversion or inversion-intercalation we can see that its around 200 which is the spected average.

Third Conclusion:
Hypothesis:
Finally, we have the cutting, we dont know how the cutting will affect the other two groups (insertions – inversions and intercalations).

Conclusion:
insertion-cut, cut-insertion is the best performance in N=32 (Generation 150) with 300 points
invertion-cut, cut-invertion is the middle performance in N=32 (Generation 150) with 150 points
intercalation-cut, cut-intercalation is the worst performance in N=32 (Generation 150) with 100 points.

Fourth Conclusion:
Hypothesis:
The order of operations will give similar results, example: insertion-inversion and inversion-insertion, intercalation-cut, cut-intercalation...

Conclusion:
The hypothesis is wrong.

(insert or invert) – intercalation: has the same increase rate as intercalation – (insert or invert), however, from generation 0, (insert or inver)-intercalation starts much further than intercalation-(insert or invert). The first starts from 150 points whereas the second one from 50 points.

Intercalation-cut: even if it starts in the same position in generation 0 as its opposite (cut-intercalation), the first gets around 150 points (N=32 generation 150) and the second at most 100 (N=32 generation 150).

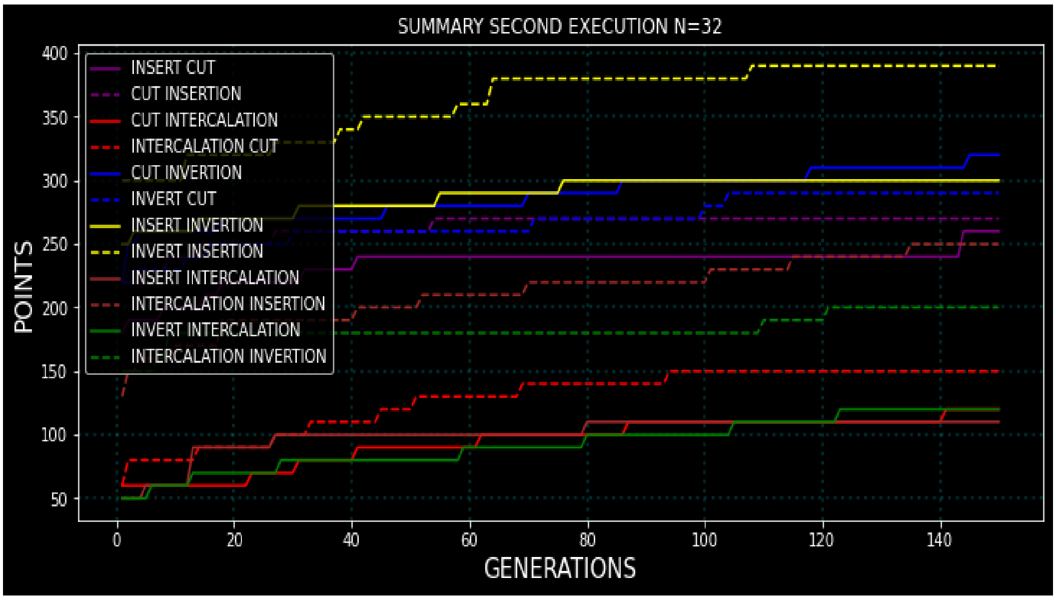
rest of the cases: They start at around 50 points of difference eachone respect to its opposite and as the increase rate is equal, at the end (N=32 generation 150) they still remain with the same difference.

Lets analyze which technique combinations are better than their opposites to try to understand what is happening (lets suppose we are at N=32 and Generation 150):
Minimum Scores:
INSERT CUT: 250
INVERT CUT:300
INSERT INVERTION:300
INSERT INTERCALATION:100
INVERT INTERCALATION:130
CUT INTERCALATION:120
Maximum Scores: (Opposites)
CUT INSERTION: 275
CUT INVERTION:350
INVERT INSERTION:400
INTERCALATION INSERTION:250
INTERCALATION INVERTION:200
INTERCALATION CUT:150

We can clearly see that a pattern is repeating in the data: when intercalation goes first it performs better than going second, and when insertion goes first it performs better than when going first. Why may be this the case?

Intercalation Hypothesis:
We can see that in generation 0 actually the values are completely different, and then through generations the increase rate remains the same in both cases, thus, we can say that just while just generating the random trees it finds better solutions when intercalation goes as the first technique of the combination. Why? Ideiarik ez.

Insertion Hypothesis:
Again we can see that in generation 0 actually the values are completely different, and then through generations the increase rate remains the same in both cases, thus, we can say that just while just generating the random trees it finds better solutions when insertion goes as the second technique of the combination. Why? Ideiarik ez.



<u>General Conclusions of Test 2:</u>
First, we can clearly see that combining insertions and inversions have the same performance of using just insertions or just inversions.

Second, while combining intercalation with insertions or inversions we can also see that as the performance of intercalation is bad and the performance of insertions and inversions is good, we get an average result that is an in-between performance.

Third, the cutting ez dakit.

Fourth, is quite interesting to see that the order of the techniques affects the performance. They have the same increase rate each technique group to their opposite techniques but some of them start from a higher point rate than the other, exactly:

-when intercalation goes first it performs better than going second.
-when insertion goes first it performs better than when going first.

Ez dakit zergatik oraindik

2. PART: GP EVOLUTION ANALYSIS:

FIRST TEST:
AFTER FINISHING 1.st PART.