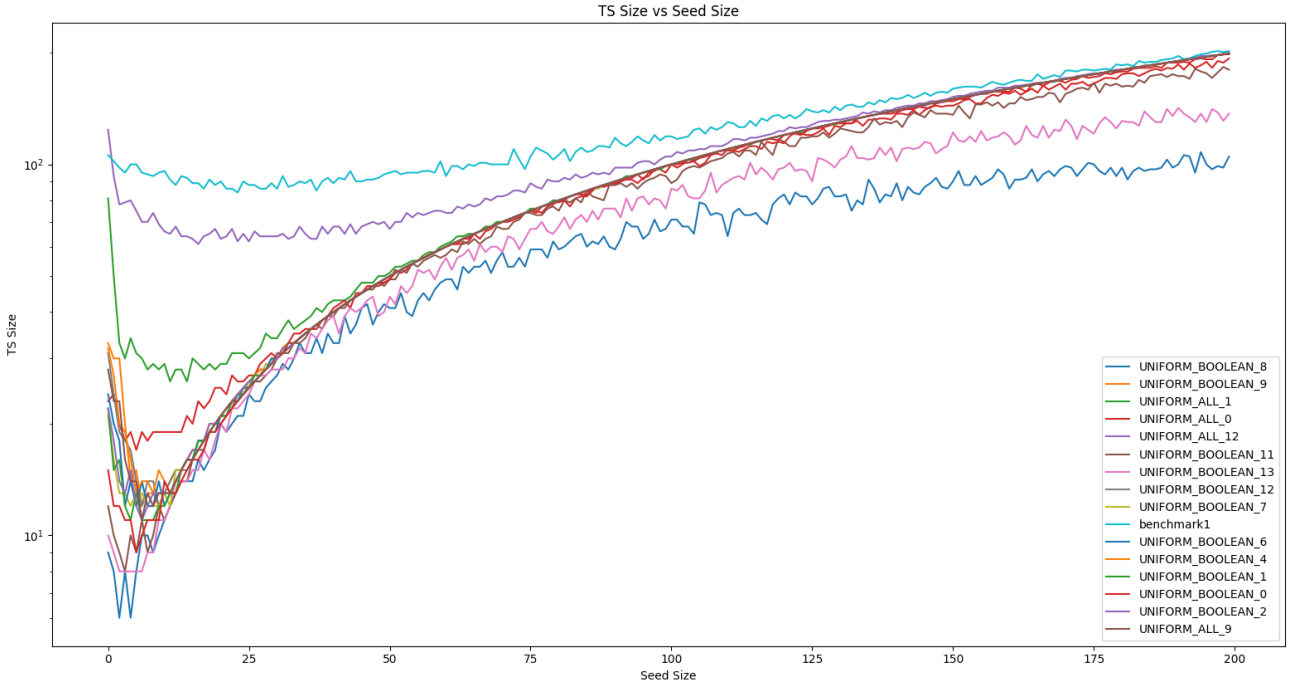
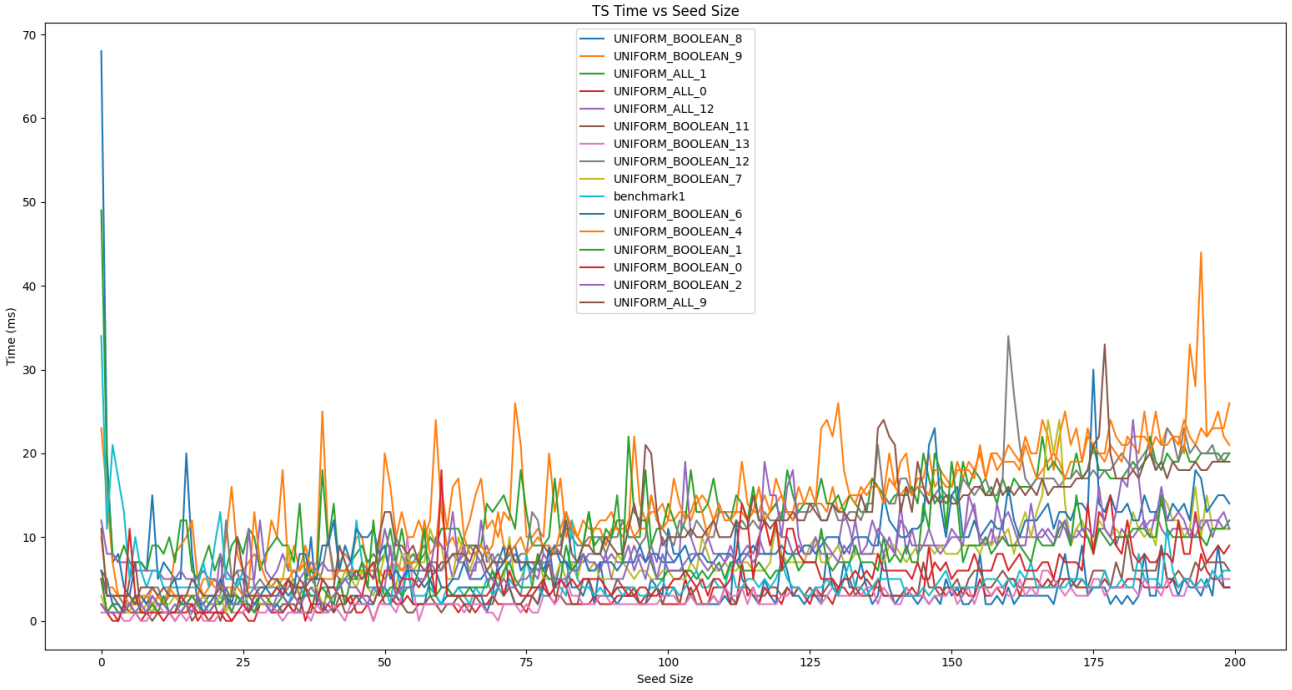
Mixing RANDOM GENERATION and pMEDICI+

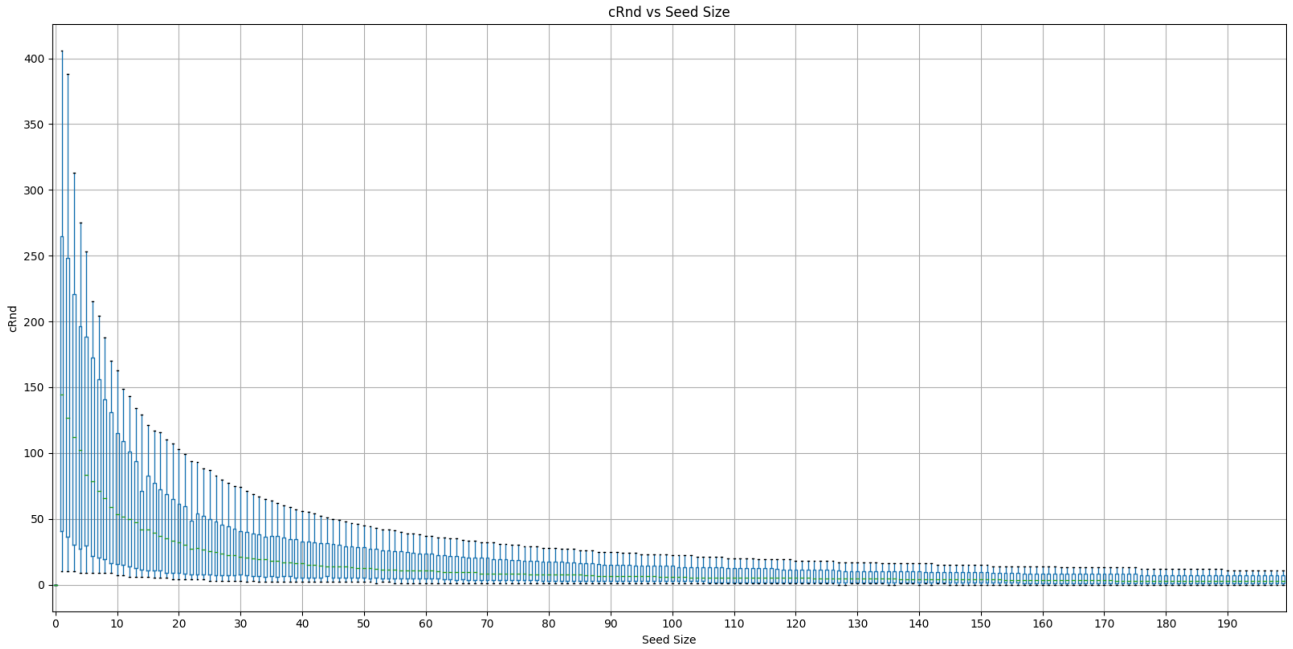
Model	Number of parameters	Alphabet size
Benchmark1	5	7
UNIFORM_ALL_0	7	3
UNIFORM_ALL_1	23	3
UNIFORM_ALL_12	12	5
UNIFORM_ALL_9	25	2
UNIFORM_BOOLEAN_0	13	2
UNIFORM_BOOLEAN_1	17	2
UNIFORM_BOOLEAN_2	19	2
UNIFORM_BOOLEAN_4	28	2
UNIFORM_BOOLEAN_6	21	2
UNIFORM_BOOLEAN_7	18	2
UNIFORM_BOOLEAN_8	7	2
UNIFORM_BOOLEAN_9	29	2
UNIFORM_BOOLEAN_11	10	2
UNIFORM_BOOLEAN_12	27	2
UNIFORM_BOOLEAN_13	8	2

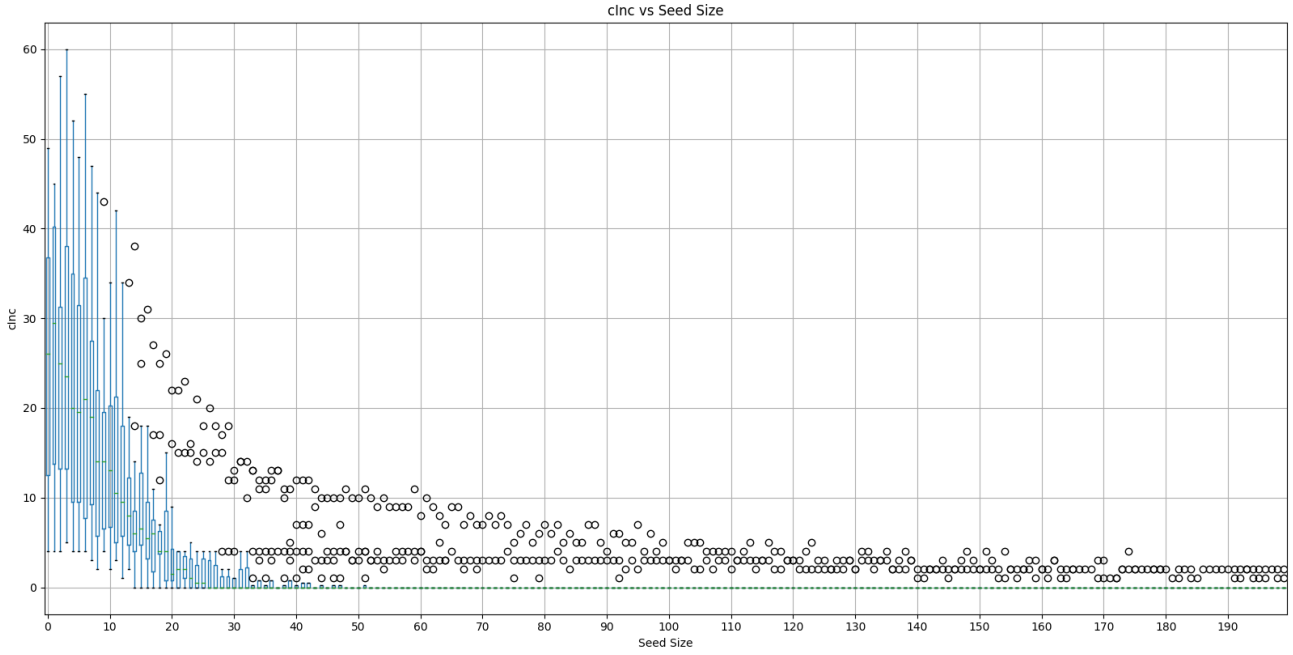
- *All models are UNIFORM*
- *This time, I kept only models with <nParams, alphabetSize> different (i.e., this is a subset of the models used in the first report)*
- *Taken from the last competition (except for Benchmark1) which is taken from the paper "One-Test-at-a-Time Heuristic Search for Interaction Test Suites"*
- *Seed sizes from 0 to 190 (step 1)*
- *Test suites generated with pMEDICI+ using only one thread in order to evaluate the impact of test seeding and exclude that of multithreading*
- *Additionally, I measured the average number of tuples covered by each test (both by the random part and by the incremental part)*

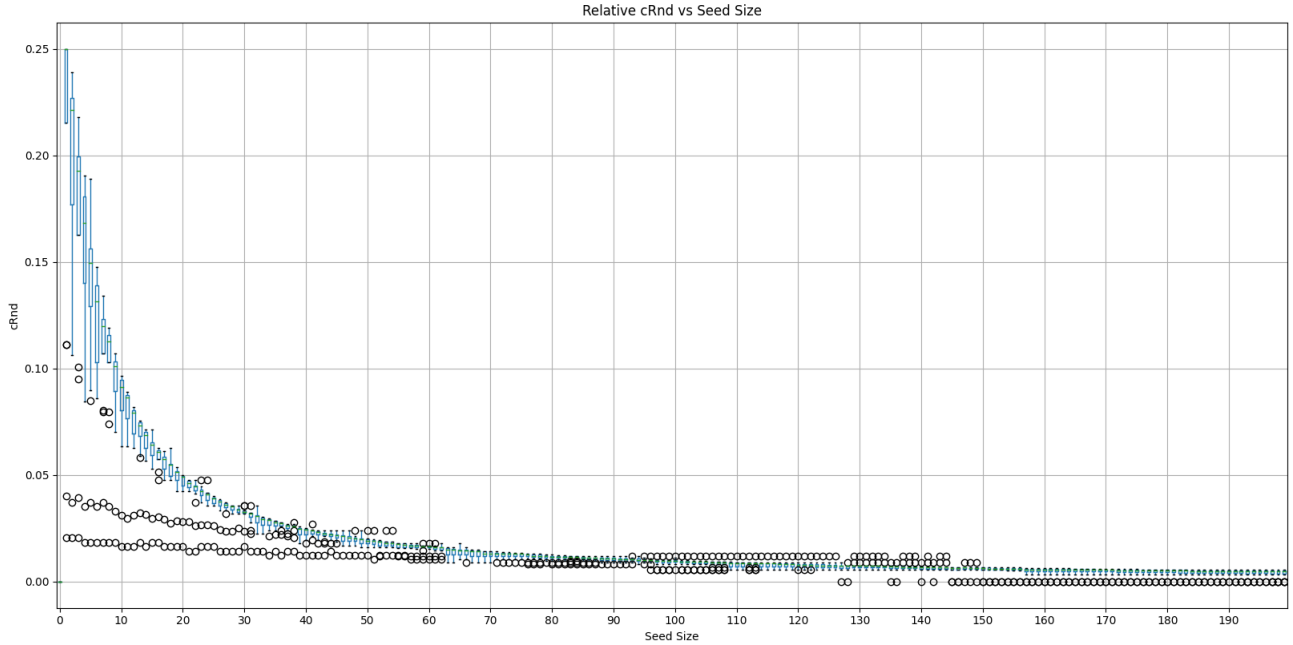
STRENGTH t=2

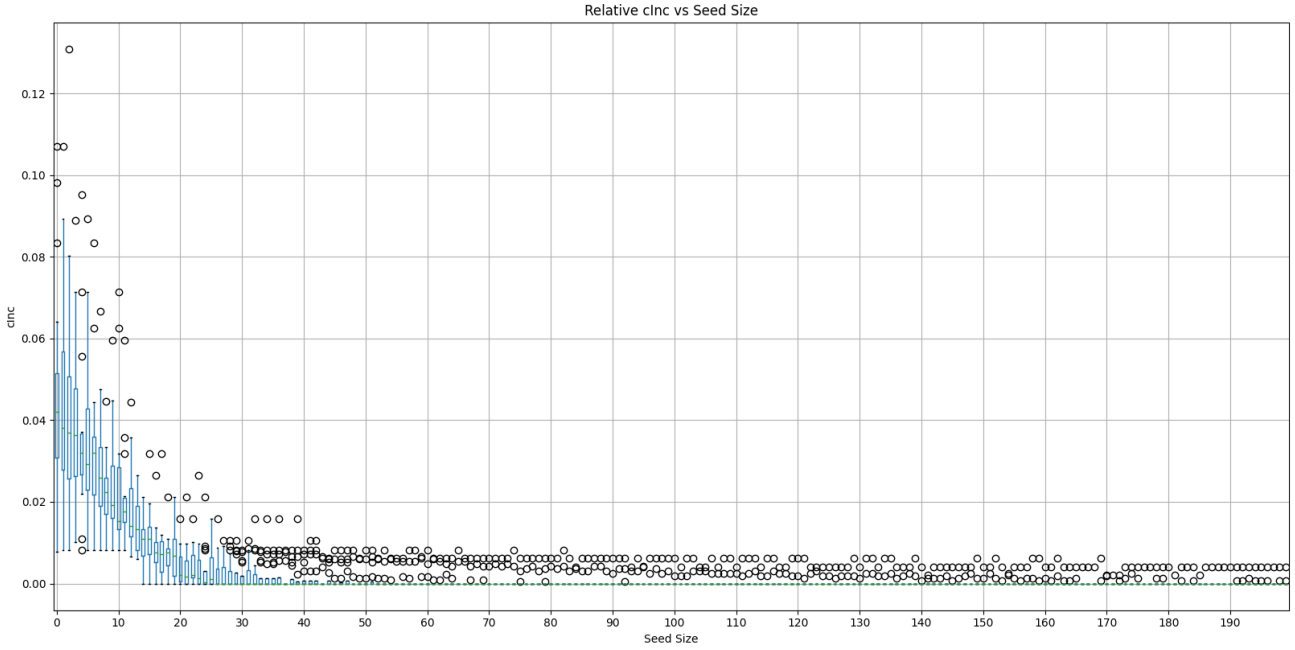

TS Size vs Seed Size
TS Size
Seed Size
10^2
10^1
0
25
50
75
100
125
150
175
200
UNIFORM_BOOLEAN_8
UNIFORM_BOOLEAN_9
UNIFORM_ALL_1
UNIFORM_ALL_0
UNIFORM_ALL_12
UNIFORM_BOOLEAN_11
UNIFORM_BOOLEAN_13
UNIFORM_BOOLEAN_12
UNIFORM_BOOLEAN_7
benchmark1
UNIFORM_BOOLEAN_6
UNIFORM_BOOLEAN_4
UNIFORM_BOOLEAN_1
UNIFORM_BOOLEAN_0
UNIFORM_BOOLEAN_2
UNIFORM_ALL_9


TS Time vs Seed Size
Time (ms)
Seed Size
0
10
20
30
40
50
60
70
0
25
50
75
100
125
150
175
200
UNIFORM_BOOLEAN_8
UNIFORM_BOOLEAN_9
UNIFORM_ALL_1
UNIFORM_ALL_0
UNIFORM_ALL_12
UNIFORM_BOOLEAN_11
UNIFORM_BOOLEAN_13
UNIFORM_BOOLEAN_12
UNIFORM_BOOLEAN_7
benchmark1
UNIFORM_BOOLEAN_6
UNIFORM_BOOLEAN_4
UNIFORM_BOOLEAN_1
UNIFORM_BOOLEAN_0
UNIFORM_BOOLEAN_2
UNIFORM_ALL_9

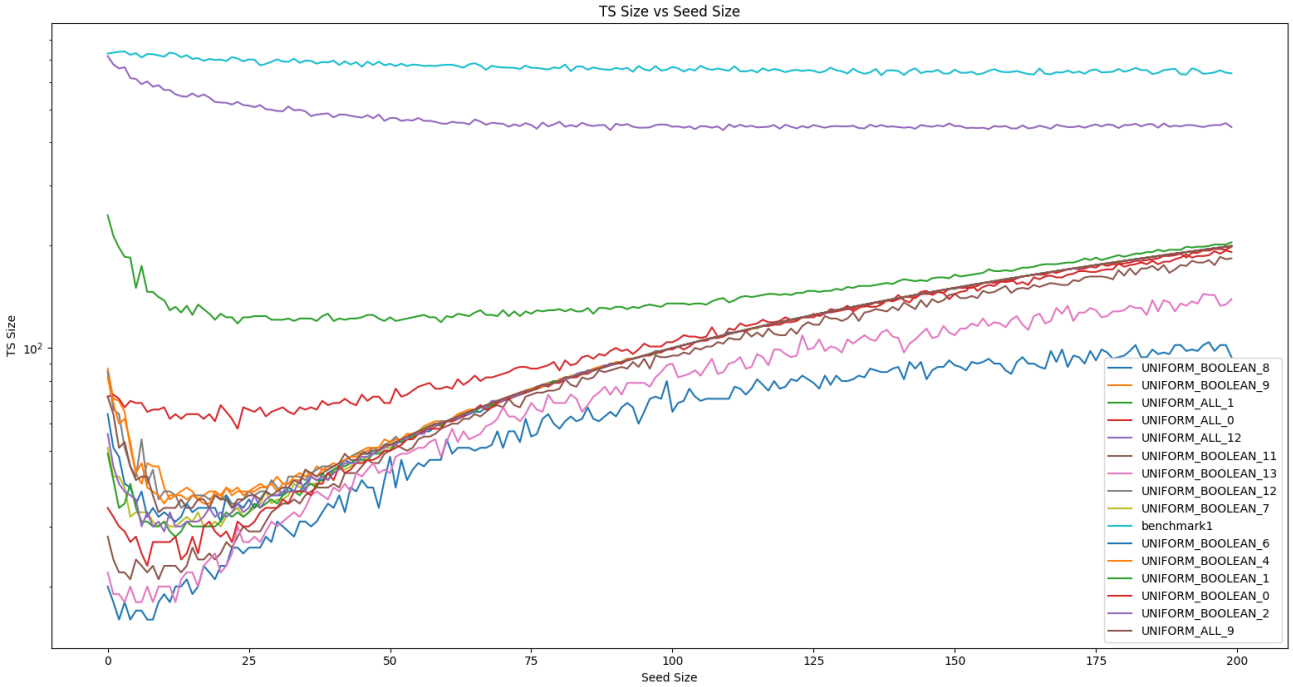

cRnd vs Seed Size
cRnd
Seed Size
0
50
100
150
200
250
300
350
400
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190

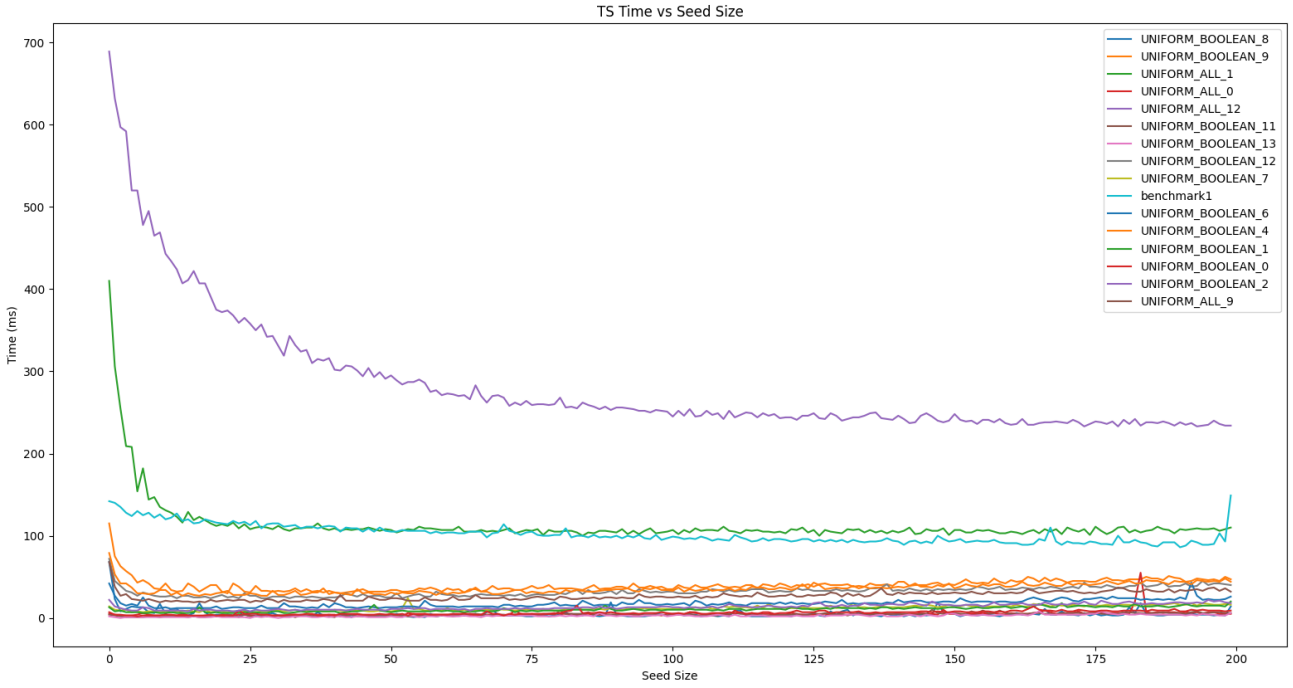

cInc vs Seed Size
cInc
Seed Size
0
10
20
30
40
50
60
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190

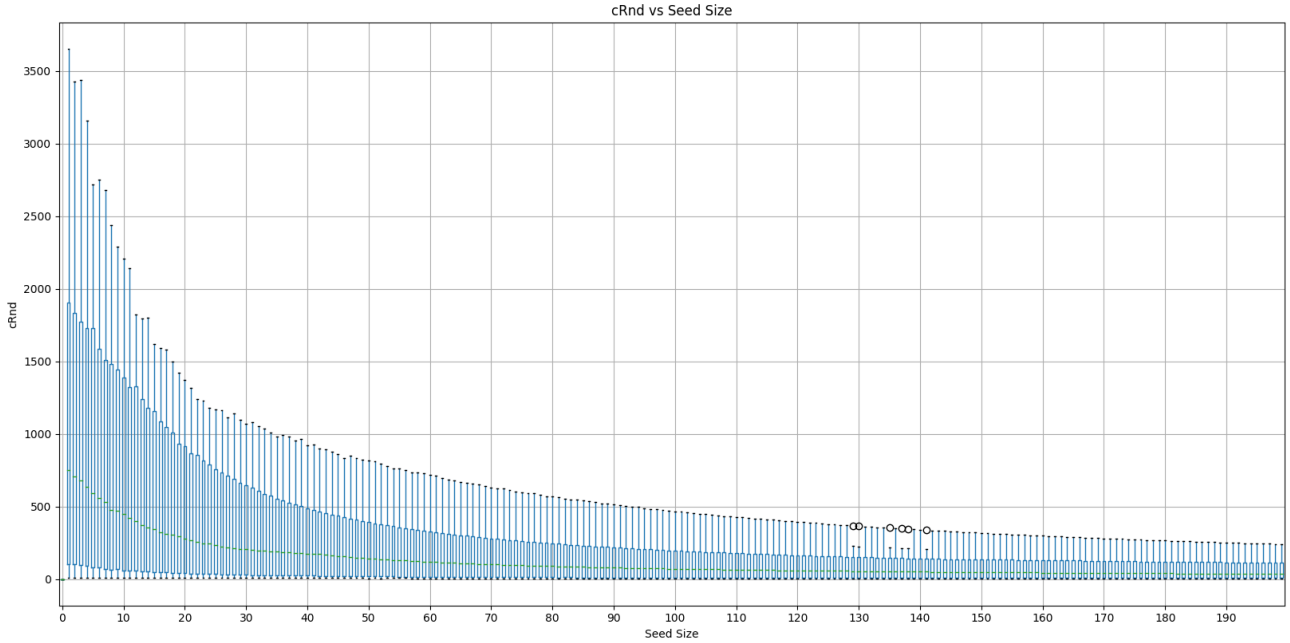

Relative cRnd vs Seed Size
cRnd
Seed Size

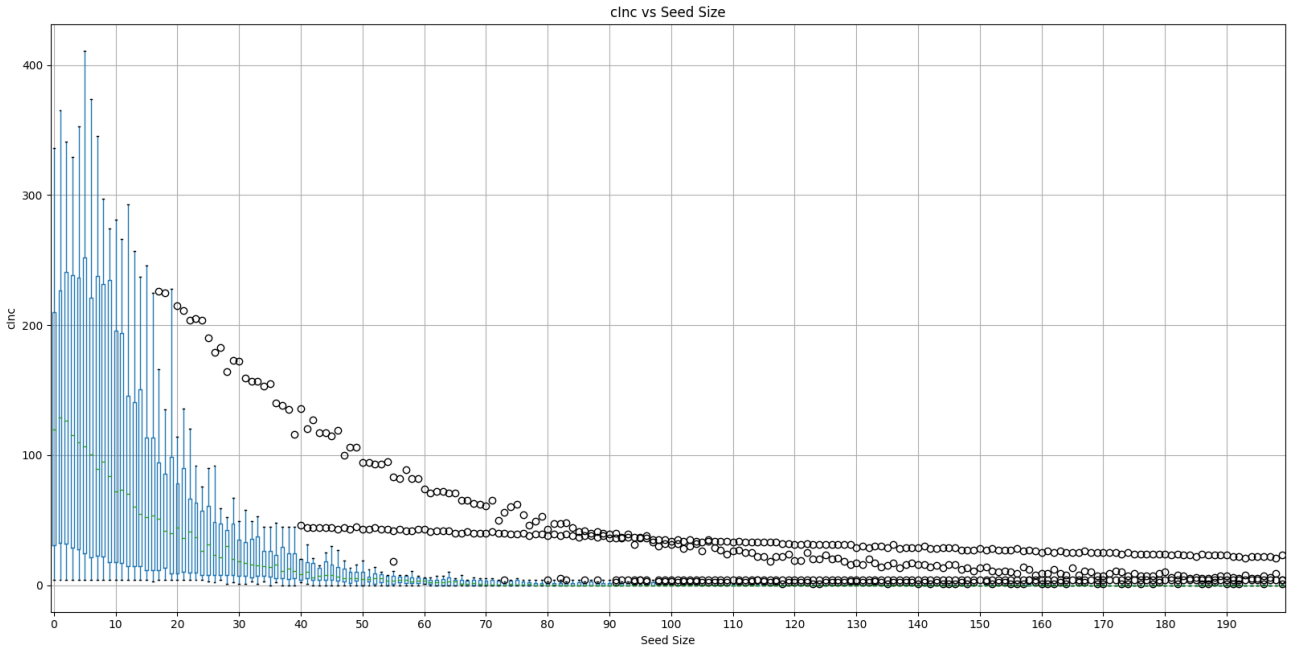

Relative cInc vs Seed Size
cInc
Seed Size

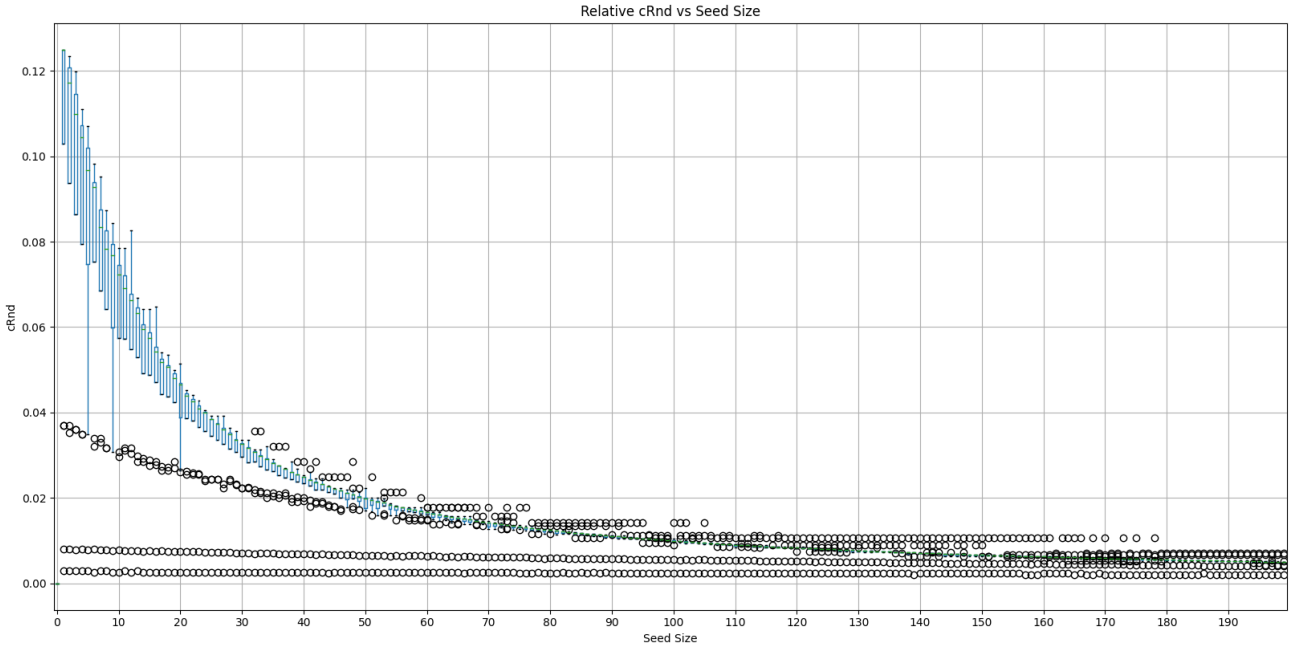
STRENGTH t=3

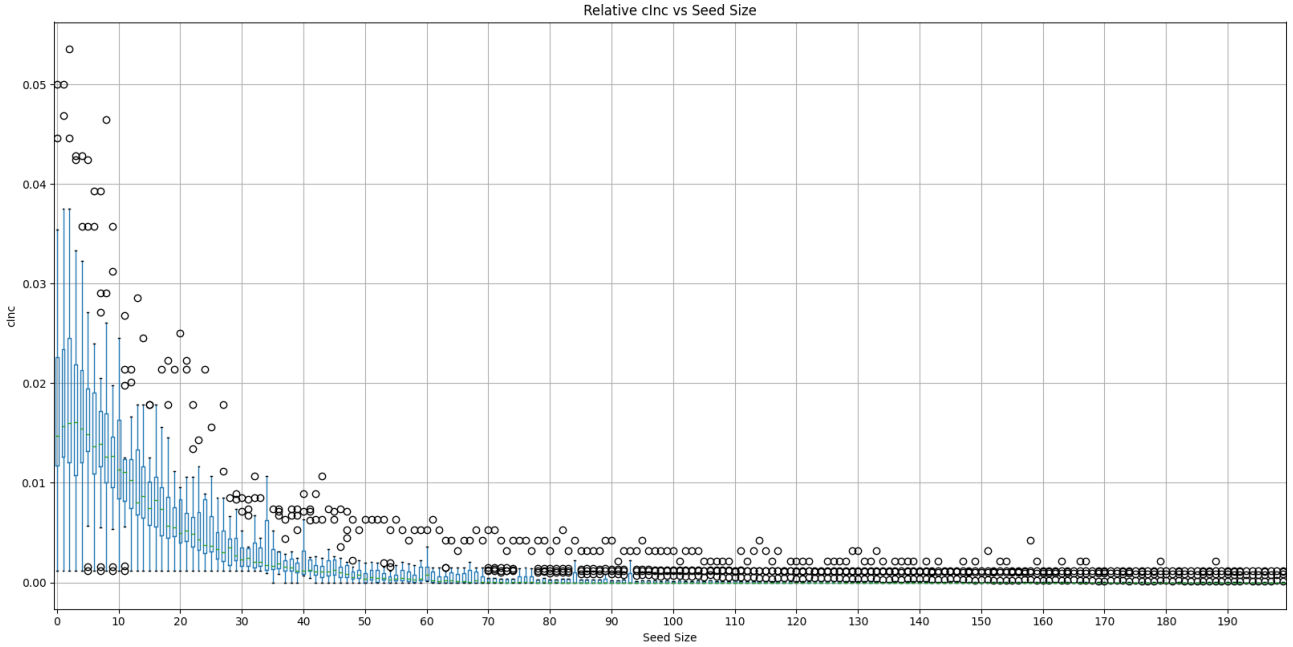

TS Size vs Seed Size
TS Size
10^2
Seed Size
0
25
50
75
100
125
150
175
200
UNIFORM_BOOLEAN_8
UNIFORM_BOOLEAN_9
UNIFORM_ALL_1
UNIFORM_ALL_0
UNIFORM_ALL_12
UNIFORM_BOOLEAN_11
UNIFORM_BOOLEAN_13
UNIFORM_BOOLEAN_12
UNIFORM_BOOLEAN_7
benchmark1
UNIFORM_BOOLEAN_6
UNIFORM_BOOLEAN_4
UNIFORM_BOOLEAN_1
UNIFORM_BOOLEAN_0
UNIFORM_BOOLEAN_2
UNIFORM_ALL_9


TS Time vs Seed Size
Time (ms)
0
100
200
300
400
500
600
700
Seed Size
0
25
50
75
100
125
150
175
200
UNIFORM_BOOLEAN_8
UNIFORM_BOOLEAN_9
UNIFORM_ALL_1
UNIFORM_ALL_0
UNIFORM_ALL_12
UNIFORM_BOOLEAN_11
UNIFORM_BOOLEAN_13
UNIFORM_BOOLEAN_12
UNIFORM_BOOLEAN_7
benchmark1
UNIFORM_BOOLEAN_6
UNIFORM_BOOLEAN_4
UNIFORM_BOOLEAN_1
UNIFORM_BOOLEAN_0
UNIFORM_BOOLEAN_2
UNIFORM_ALL_9

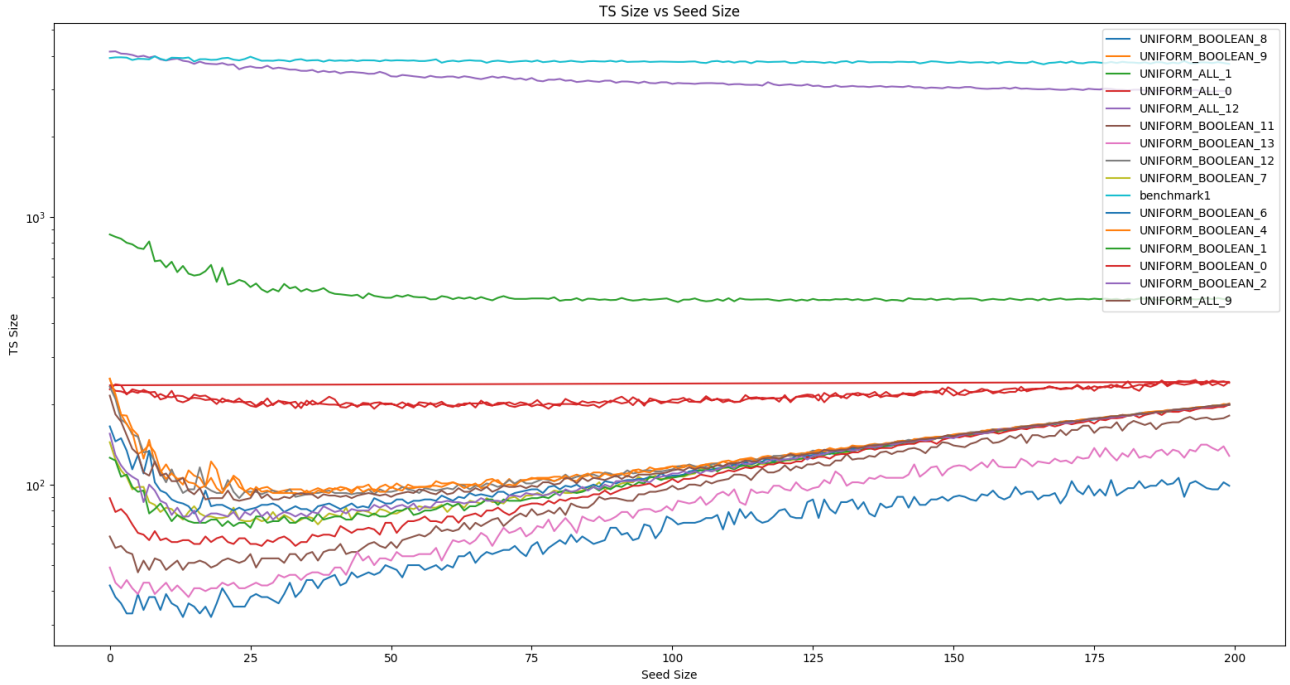

cRnd vs Seed Size
cRnd
Seed Size
0
500
1000
1500
2000
2500
3000
3500
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190

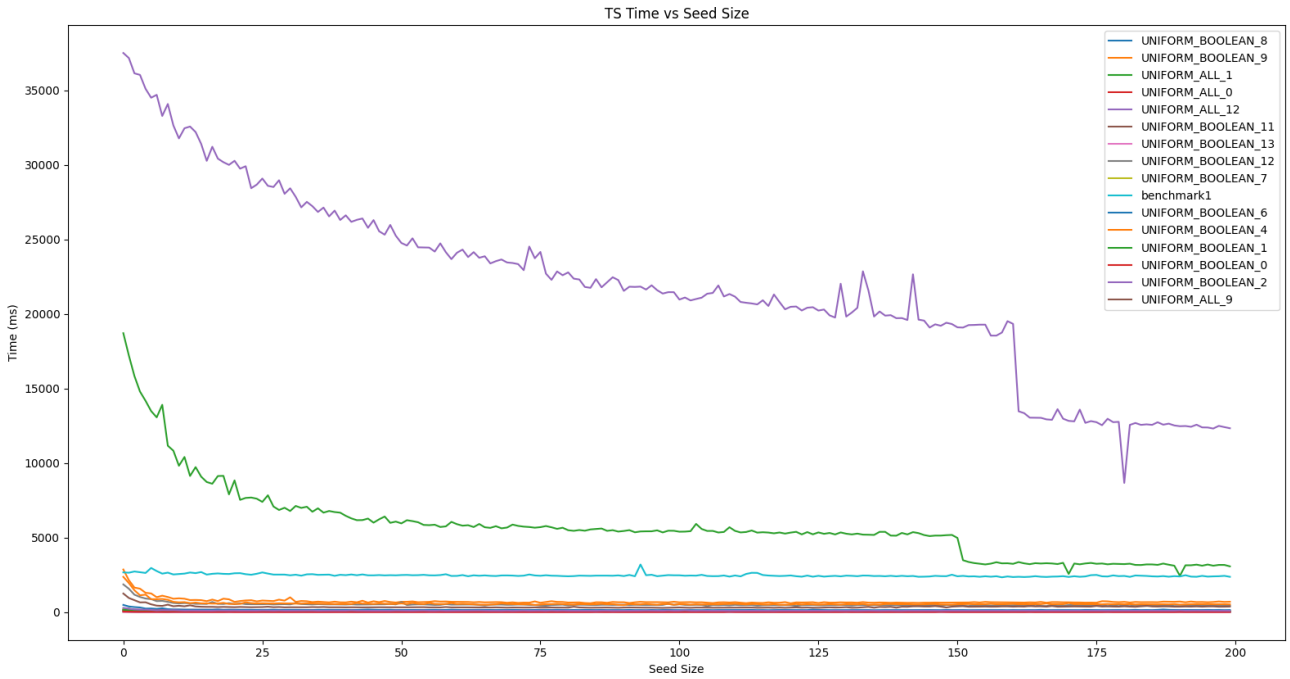

cInc vs Seed Size
cInc
Seed Size
0
100
200
300
400
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190


Relative cRnd vs Seed Size
cRnd
Seed Size

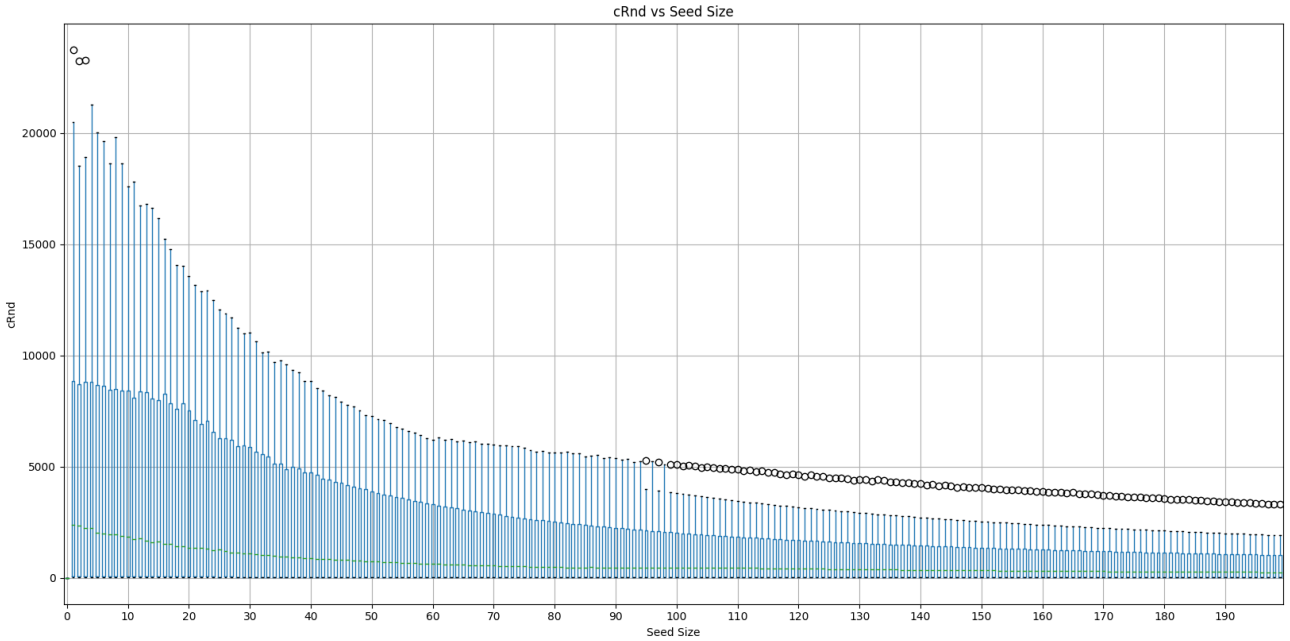

Relative cInc vs Seed Size
cInc
Seed Size

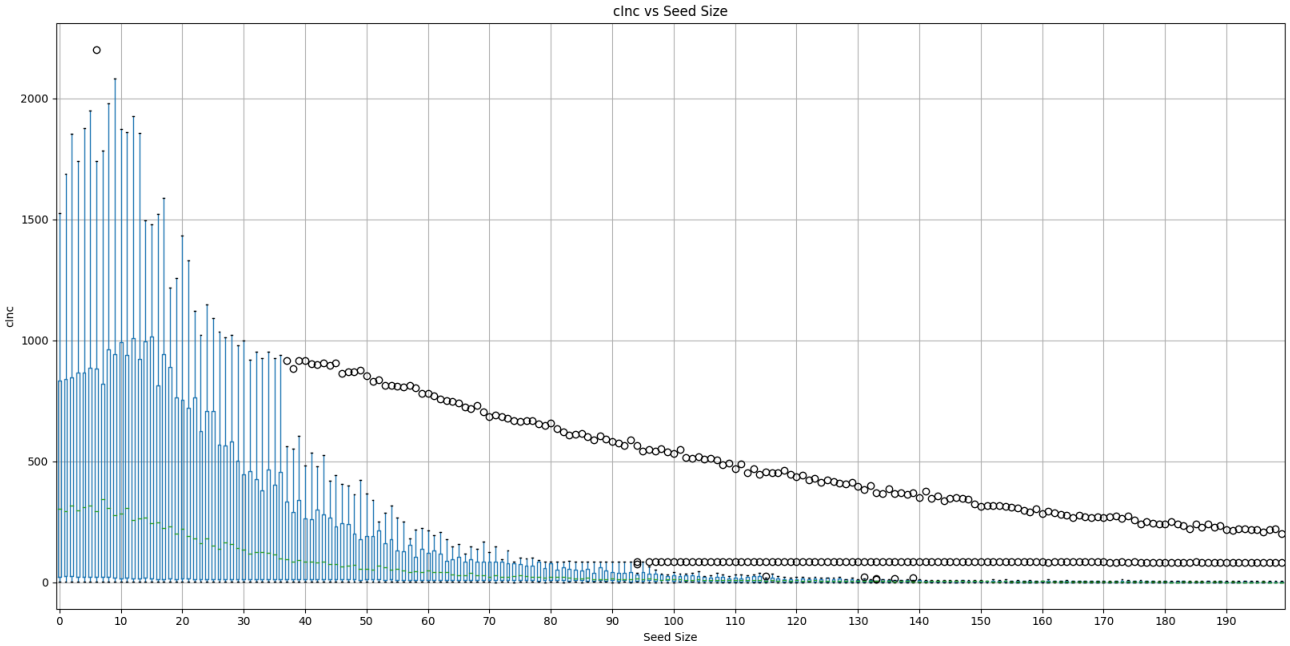
STRENGTH t=4

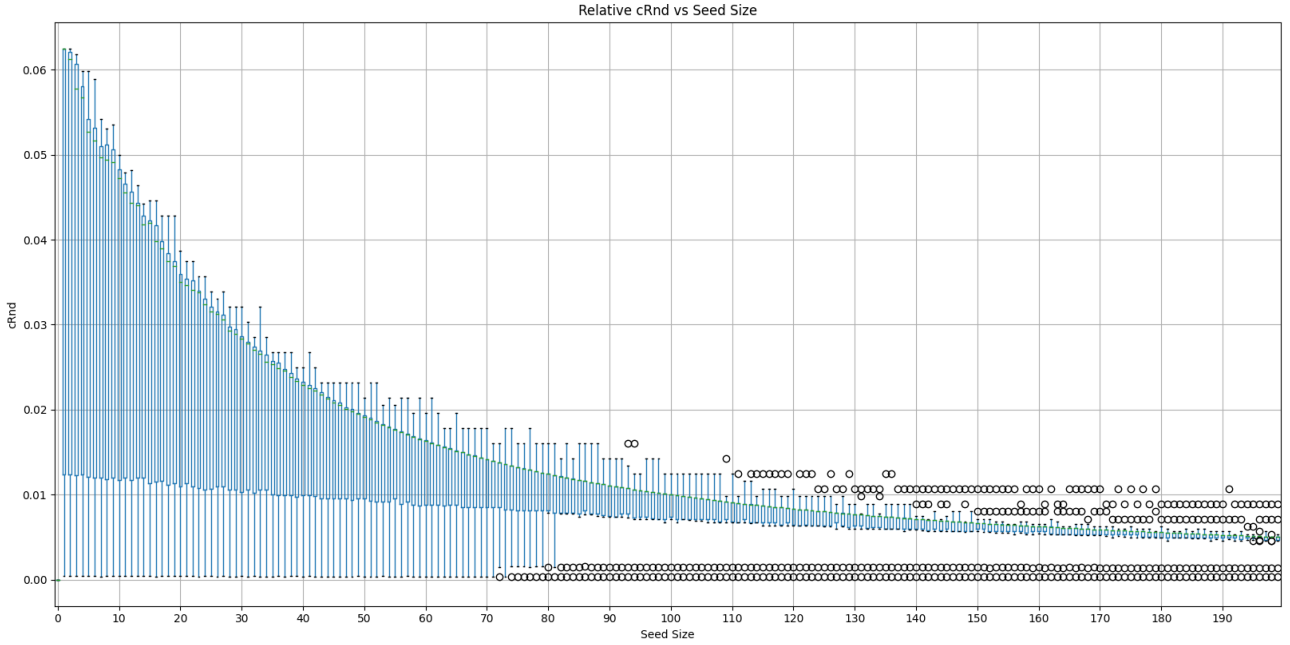


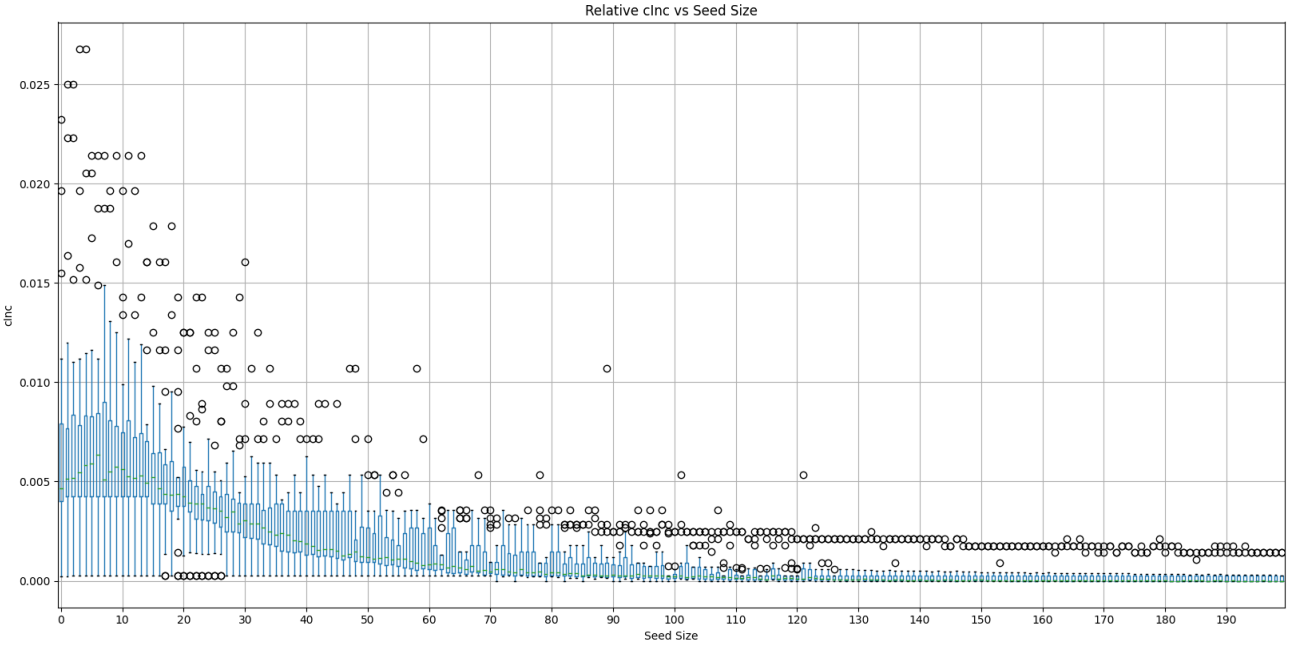


Note that my PC turned off while generating tests for the two most complex models, that's why we have that reduction after 150 and 160 seeds.


cRnd vs Seed Size
cRnd
Seed Size
0
5000
10000
15000
20000
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190


cInc vs Seed Size
cInc
Seed Size
0
500
1000
1500
2000
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190


Relative cRnd vs Seed Size
cRnd
Seed Size


Relative cInc vs Seed Size
cInc
Seed Size

SOME CONSIDERATION

- In general there is an advantage in terms of generation time when increasing the seed size. The higher the model complexity, the higher the advantage.

- In terms of size, it depends on the number of possible test cases. There is a point beyond which it is useless (I would say counterproductive) to increase the seed size, since there are more random tests than those that would have been generated by the generator starting from scratch.

- It seems that the “best” point for the test suite size is also very near the “best” point for the generation time.

- Especially for higher strengths the advantage of including a random pre-generation in complex models is very significant, in particular in terms of time.

- The number of average covered tuples per test case is not easily generalizable, since it depends on the number of parameters that the considered IPM has. For example, if we consider a model with only a couple of parameters, only a single tuple can be covered by each test case, and it is not due to the algorithm we are using, but only to the model itself.
  For this reason, I have computed a relative measure, dividing the c of the algorithm by the total number of tuples. However, I’m not sure of how we can use it.

- For the random part, if one increases the number of seeds, in general, the c decreases since we add more test cases but the number of previously-uncovered tuples that are now covered decreases. The same is for the incremental part, as the higher the size of the seeds, the higher the number of tuples that are covered by the random part, and the lower the number of tuples to be covered by the incremental part.