Mixing Random Generation and pMEDICI+

New Activities

1) I generated 30 new models, with higher variability in terms of k (2 <= k <= 30) and v (2 <= v <= 7). I attach the new models as an archive to the email.

2) I repeated the experiments for t=2 and t=3 (even if for t=3, some instances timed out) and produced some plots by ordering the instances by v.

3) I investigated which was the percentage of tuples covered by random tests when the minimum size/time has been found.

4) I tried to define which should have been the minimum size/time if the number of seeds obtained with the formula you proposed has been used and compared it to the actual min. Moreover, I compared the size/time with that of the baseline approach (i.e., with seedSize = 0)

5) For strength t=2, I tried executing each tuple <model, seedsize> 100 times and I repeated the experiments in points 2, 3, and 4.

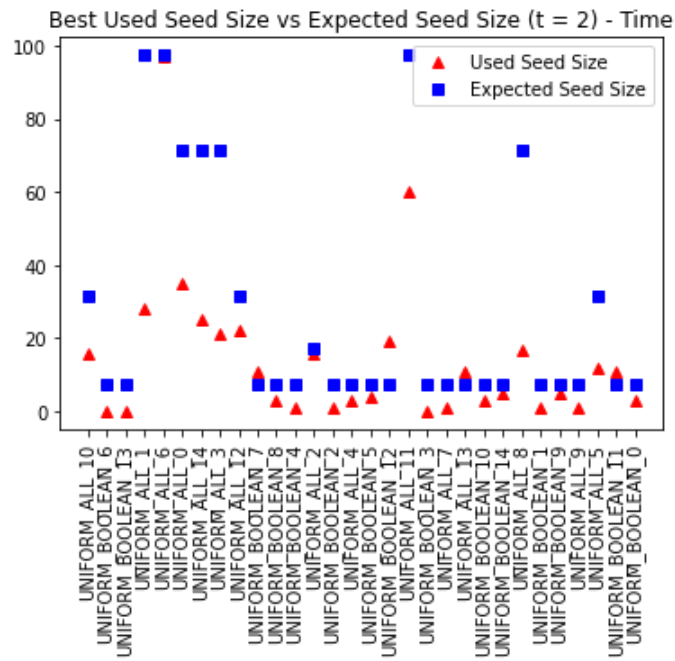
General points

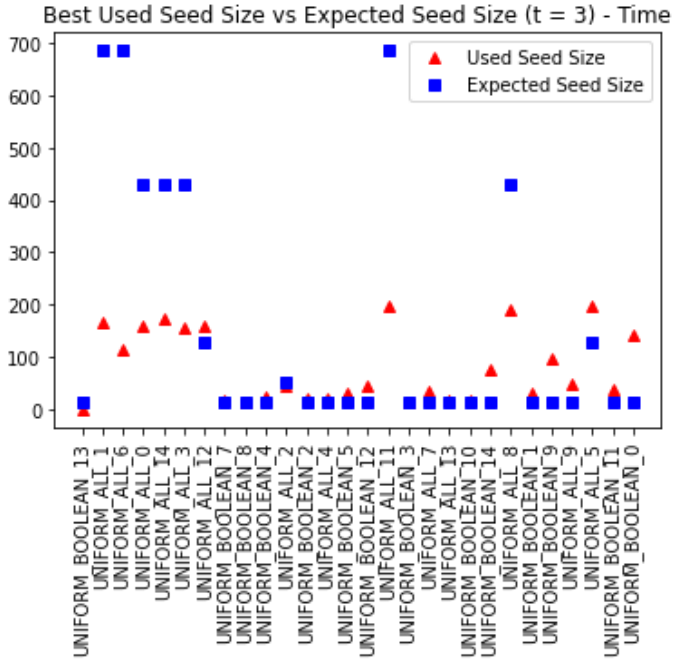
- There are some models with k=2 (UNIFORM_BOOLEAN_6, UNIFORM_ALL_10), thus they are not considered for strength t=3.

Point n. 2

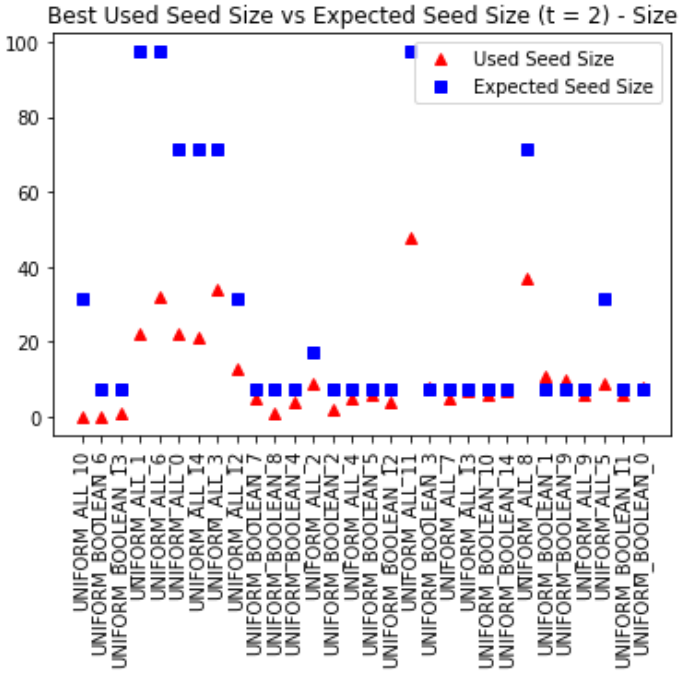
When repeating the experiments on the new models, the following plots showing the best used seed size (obtained with the experiments) compared to that forecasted by the formula have been obtained:

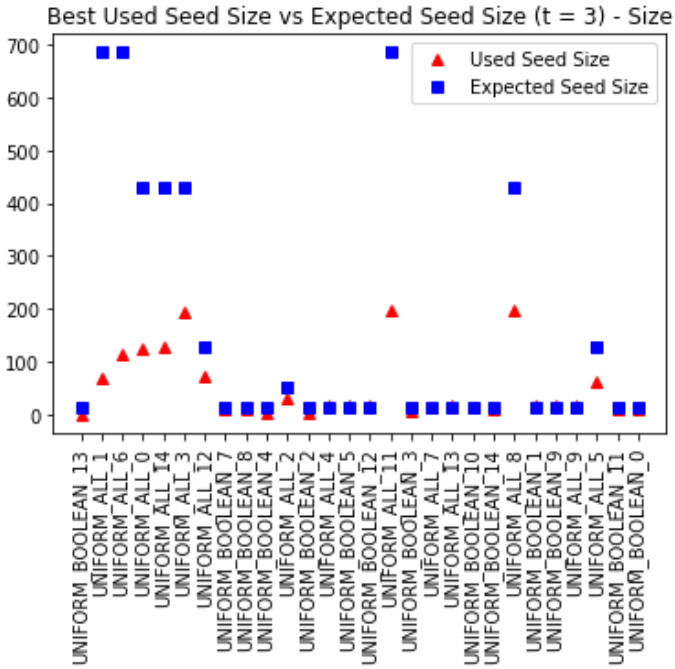
- CONSIDERING THE SCENARIO IN WHICH THE MINIMUM GENERATION TIME HAS BEEN OBTAINED





- CONSIDERING THE SCENARIO IN WHICH THE MINIMUM TEST SUITE SIZE HAS BEEN OBTAINED





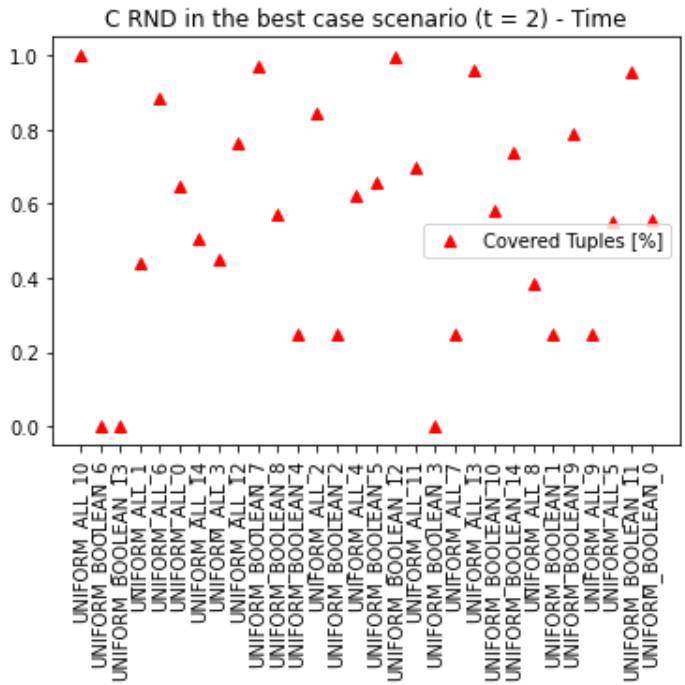
Some considerations

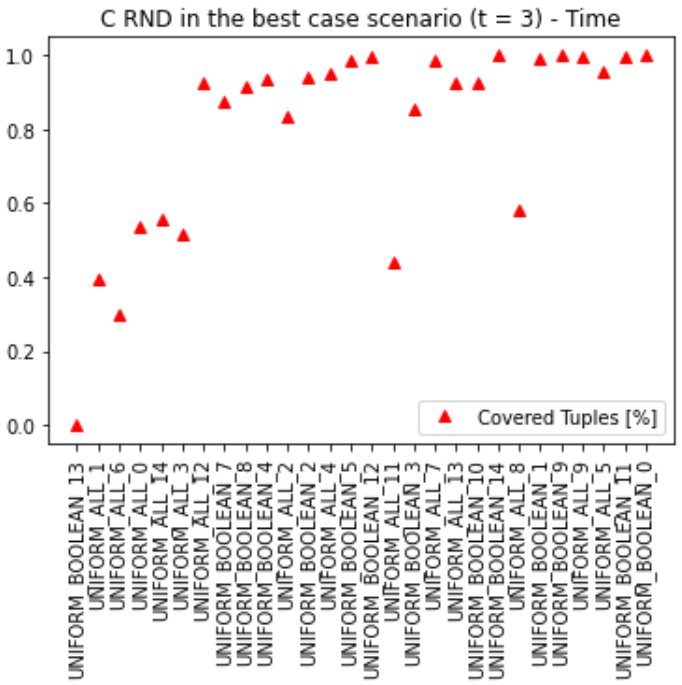
It seems that the behavior is similar to what we had last time. For some instances, the prediction fits the value we measure, while the difference is quite high for others. As last time, the higher the strength, the higher the prediction accuracy.

Point n. 3

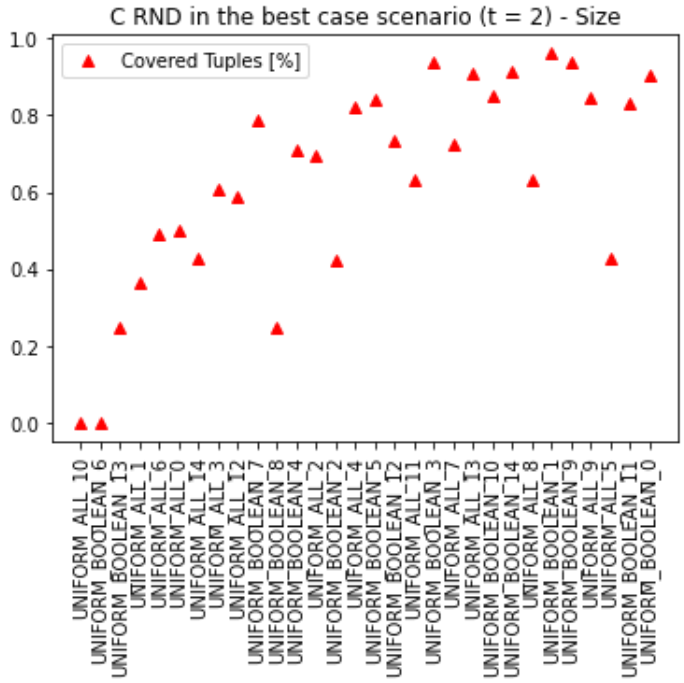
Given that I store the c coefficient for each test generation and the total number of tuples, I computed the number of tuples covered by the random part by multiplying the number of seeds (i.e., the number of random tests) for the cRnd coefficient. Then, for having all data with the same scale, I divided the obtained number by the total number of tuples of the considered instance.

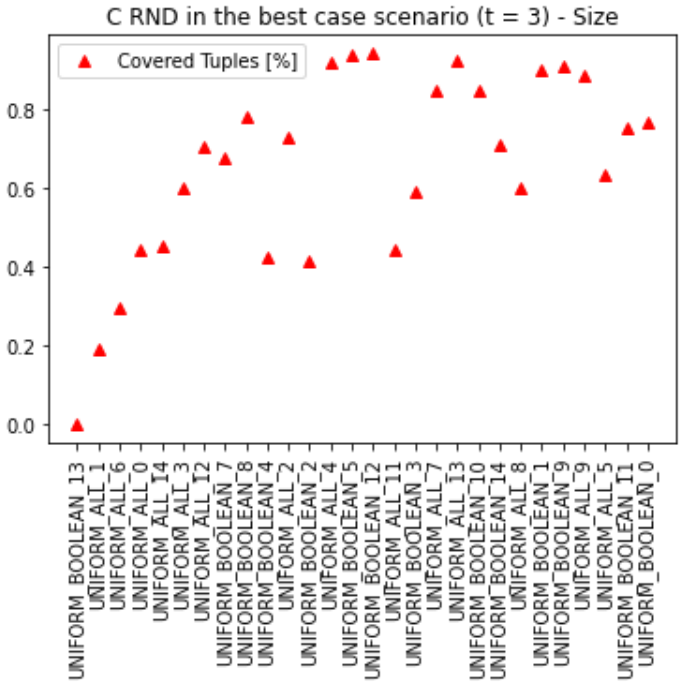
- CONSIDERING THE SCENARIO IN WHICH THE MINIMUM GENERATION TIME HAS BEEN OBTAINED





- CONSIDERING THE SCENARIO IN WHICH THE MINIMUM TEST SUITE SIZE HAS BEEN OBTAINED





Some considerations

I do not see any clear trend. Probably, it seems that when the number of parameters increases, the percentage of tuples covered by the random part increases as well (but this is somehow expected, since we have more tuples to be

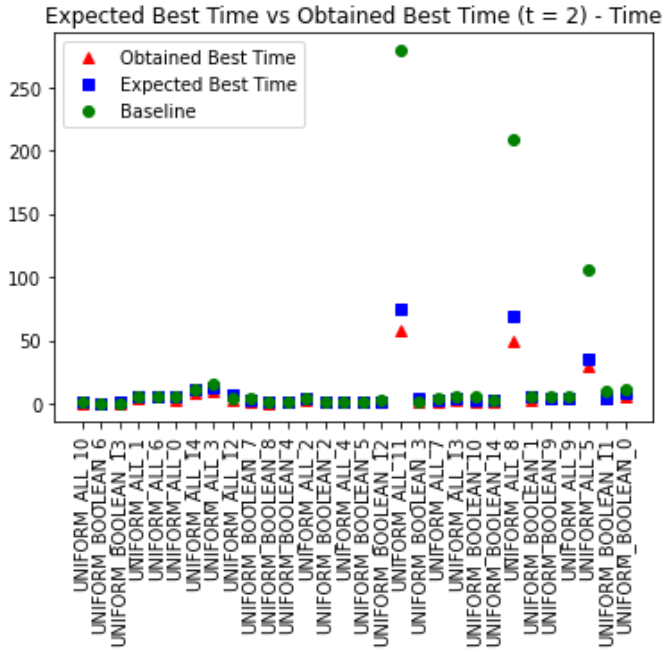
covered and it is better to cover them with the most rapid approach, i.e., the random one). In some cases, even covering 0 tuples with the random part is the best solution (e.g., UNIFORM_BOOLEAN_13 with t=3).

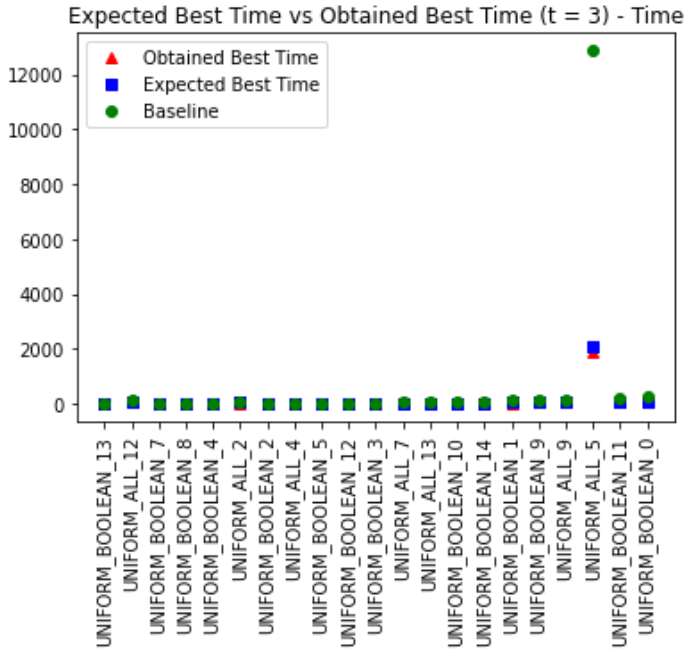
Point n. 4

At this point, I considered the seed size forecasted by the formula and that with which we obtained the best results. So, I compared the test suite size (or generation time) we had in our best configuration and what we would have obtained if the optimal predicted seed size had been used.

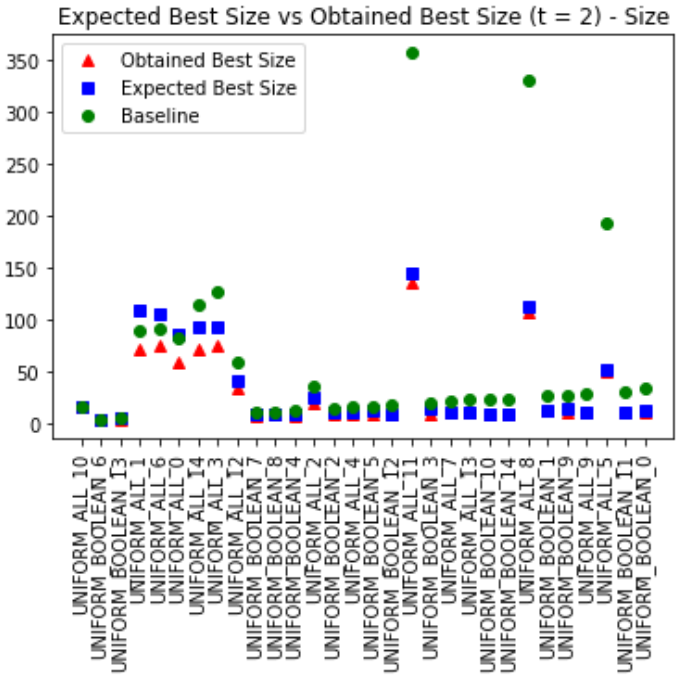
Note that it has been possible only in some cases since I executed the experiments with a maximum seed size of 200. Thus, if the formula predicted more than 200 random test cases, the comparison in that configuration cannot be executed.

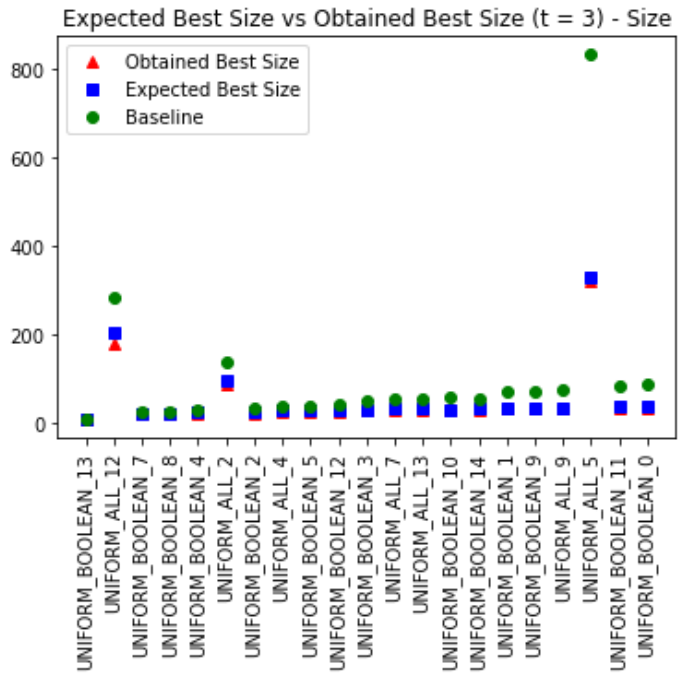
- CONSIDERING THE SCENARIO IN WHICH THE MINIMUM GENERATION TIME HAS BEEN OBTAINED





- CONSIDERING THE SCENARIO IN WHICH THE MINIMUM TEST SUITE SIZE HAS BEEN OBTAINED





Some considerations

I would say that if we only consider generation time, the seed size given by the formula gives more or less the same results we have in our best scenario. Note that on the y-axis the quantities (for time) are in milliseconds, so the difference is very small. Nevertheless, the formula always gives a number of seeds that leads to a slower generation.

As usual, the higher the strength, the better the approximation we have with the formula.

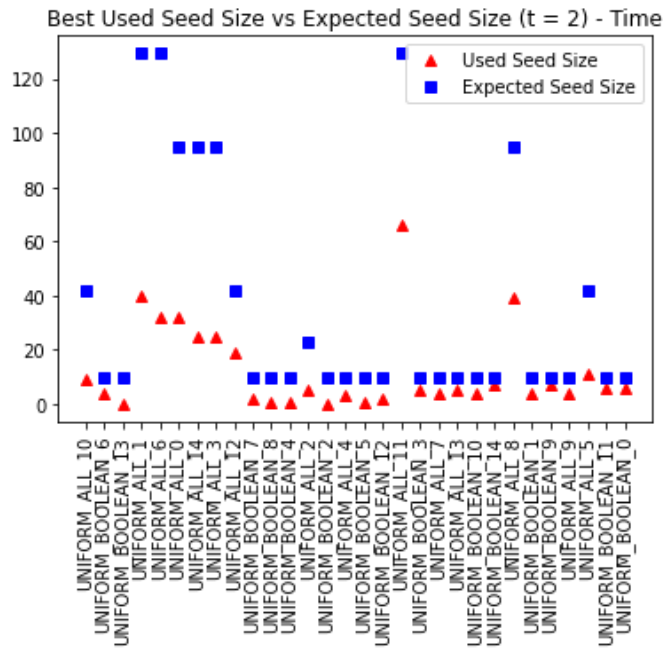
On the other hand, if we consider the test suite size, sometimes the difference is higher than what we have with time (even more than 30 test cases in the case of UNIFORM_BOOLEAN_13 for t=2).

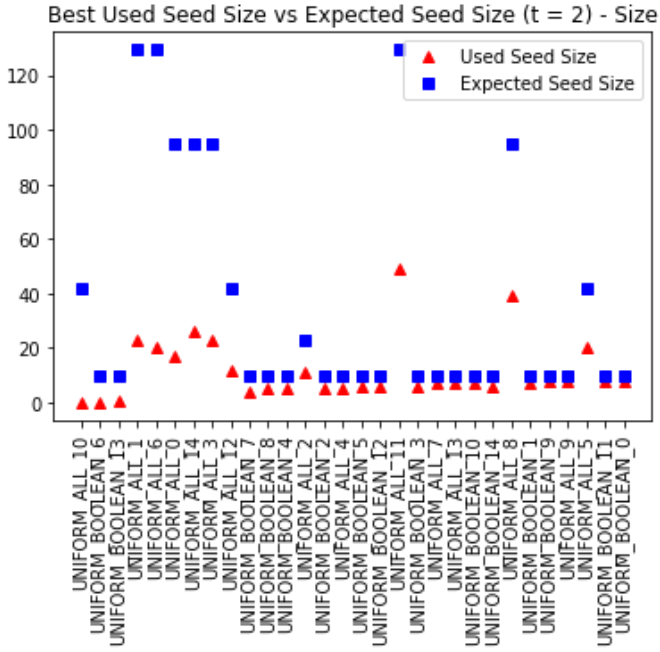
***** NICE POINT ***:** With the obtained best data (size/time) we are always performing better or equally to the baseline scenario (i.e., when only pMEDICI is applied, without the random part, with SeedSize = 0). But also looking to the results we obtain with the formula (Expected Best Size/Time) we normally perform better than the baseline except for UNIFORM_ALL_1 and UNIFORM_ALL_6 with strength t=3.

Point n.5

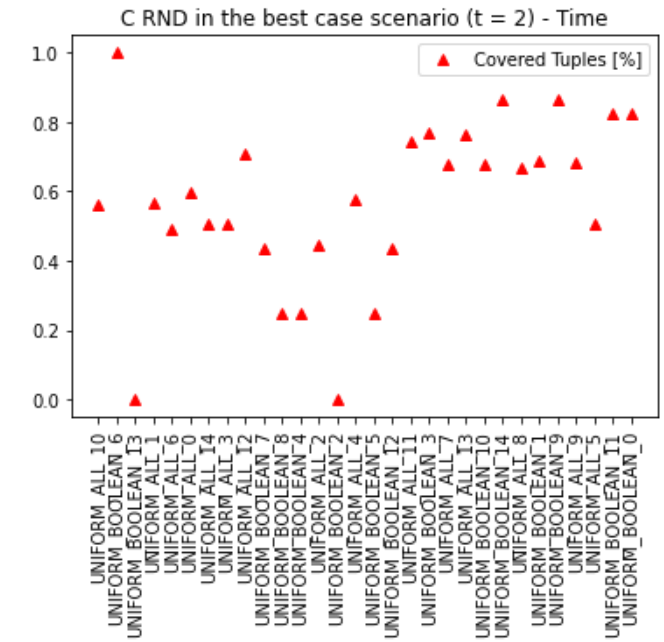
At this point, **only for strength t=2**, I repeated the previous experiments but for each tuple <model, seedsize> I repeated the experiments 100 times. In this way, we can average the randomness of each single execution.

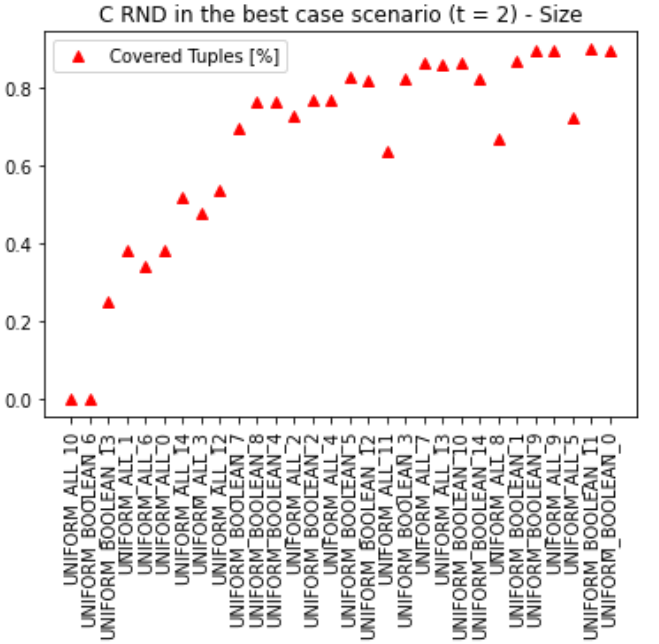
- ***POINT 2***



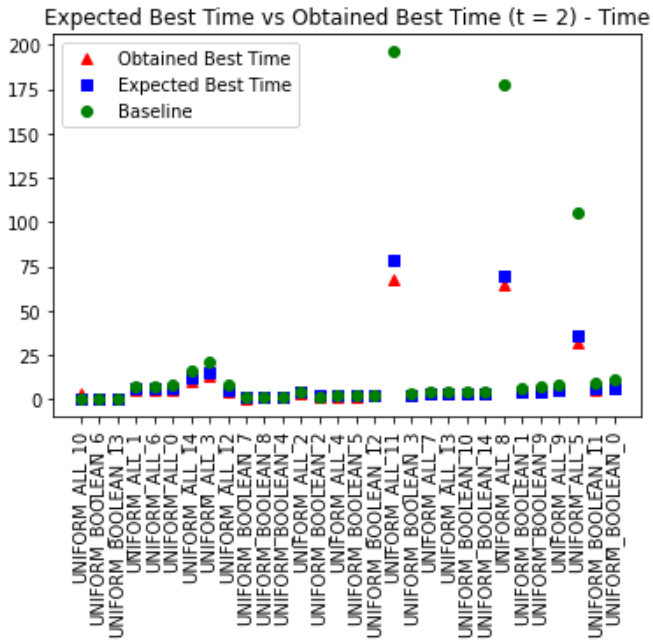


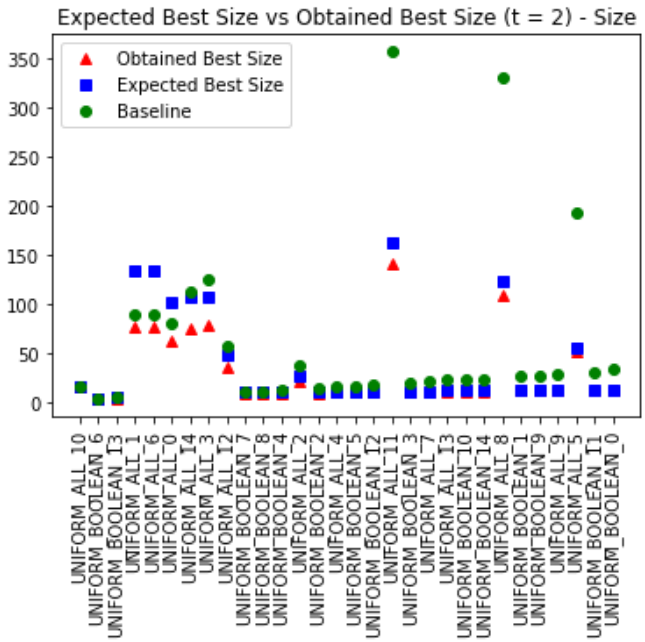
- ***POINT 3***





- ***POINT 4***





Some considerations

I would say that the considerations are the same as those I draw for points 2, 3, and 4 with a single execution.