**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***

Immagine che contiene testo, linea, Diagramma, diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Diagramma

Descrizione generata automaticamente

Immagine che contiene linea, Diagramma, diagramma, schermata

Descrizione generata automaticamenteImmagine che contiene testo, Diagramma, linea, diagramma

Descrizione generata automaticamenteImmagine che contiene linea, Diagramma, diagramma, schermata

Descrizione generata automaticamenteImmagine che contiene testo, linea, Diagramma, diagramma

Descrizione generata automaticamente

***STRENGTH t=3***

Immagine che contiene testo, linea, diagramma, Diagramma

Descrizione generata automaticamente Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente

Immagine che contiene linea, Diagramma, schermata, diagramma

Descrizione generata automaticamente Immagine che contiene linea, Diagramma, testo, diagramma

Descrizione generata automaticamente Immagine che contiene Diagramma, linea, diagramma, schermata

Descrizione generata automaticamente Immagine che contiene testo, Diagramma, diagramma, linea

Descrizione generata automaticamente

***STRENGTH t=4***

Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente

Immagine che contiene testo, linea, Diagramma, schermata

Descrizione generata automaticamente

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.

Immagine che contiene linea, Diagramma, schermata, diagramma

Descrizione generata automaticamente

Immagine che contiene Diagramma, linea, schermata, testo

Descrizione generata automaticamente

Immagine che contiene linea, Diagramma, schermata, diagramma

Descrizione generata automaticamente

Immagine che contiene testo, Diagramma, linea, diagramma

Descrizione generata automaticamente

**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.