

### Recherche de la meilleure valeur pour maxiter:

Ici je prend `nb_iter = 1000` (qui est la valeur par défaut), et `maxiter = 1000000` (qui est aussi la valeur par défaut).

- Ici j'ai testé beaucoup de codes et à chaque fois le résultat est 1
- Par contre je trouve des résultats très variés quand il s'agit de GF(2)
- Tous les codes du deuxième tableau je les testé sauf que le résultat est presque toujours 1.

<i>Code</i>	<i>Le nombre maximal de permutation pour atteindre n//k informations set</i>
<code>codes.random_linear_code(GF(2), 68, 11)</code>	7, 6, 5, 1
<code>codes.random_linear_code(GF(2), 96, 32)</code>	26, 25, 23, 1
<code>codes.random_linear_code(GF(2), 99, 33)</code>	27, 24, 23, 22, 20, 1
<code>codes.random_linear_code(GF(2), 128, 32)</code>	29, 24, 27, 23, 21, 1
<code>codes.random_linear_code(GF(3), 60, 15)</code>	10, 1
<code>codes.random_linear_code(GF(3), 64, 16)</code>	14, 12, 11, 1
<code>codes.random_linear_code(GF(3), 68, 17)</code>	11, 10, 8, 1
<code>codes.random_linear_code(GF(7), 36, 9)</code>	6, 5, 1
<code>codes.random_linear_code(GF(7), 40, 10)</code>	1
<code>codes.random_linear_code(GF(8), 36, 9)</code>	8, 1
<code>codes.random_linear_code(GF(8), 40, 10)</code>	5, 4, 1
<code>codes.random_linear_code(GF(9), 32, 8)</code>	5, 1
<code>codes.random_linear_code(GF(7**2), 35, 6)</code>	1
<code>codes.random_linear_code(GF(2), 128, 32)</code>	28, 27, 24, 23, 22, 20, 19, 1
<code>codes.random_linear_code(GF(2), 96, 32)</code>	<b>37</b> , 24, 23, 21, 19, 1
<code>codes.random_linear_code(GF(3), 96, 32)</code>	12, 10, 9, 1
<code>codes.random_linear_code(GF(13), 96, 40)</code>	1
<code>codes.random_linear_code(GF(5), 96, 32)</code>	7, 1
<code>codes.random_linear_code(GF(5), 44, 11)</code>	7, 1
<code>codes.random_linear_code(GF(2), 111, 11)</code>	12, 1

<code>codes.random_linear_code(GF(3),99,33)</code>	13, 1
<code>codes.random_linear_code(GF(7),111,33)</code>	1
<code>codes.random_linear_code(GF(5),48,12)</code>	8, 6, 1
<code>codes.random_linear_code(GF(2),60,14)</code>	4, 1,
<code>codes.random_linear_code(GF(2),100,14)</code>	7, 6, 1
<code>codes.random_linear_code(GF(2),100,20)</code>	29, 26, 24, 21, 18, 1
<code>codes.random_linear_code(GF(3),100,20)</code>	11, 10, 9, 1
<code>codes.random_linear_code(GF(4),100,20)</code>	10, 9, 7, 1
<code>codes.random_linear_code(GF(5),100,25)</code>	7, 1
<code>codes.random_linear_code(GF(5),95,25)</code>	1
<code>codes.random_linear_code(GF(2),195,13)</code>	30, 25, 21, 20, 1
<code>codes.random_linear_code(GF(3),195,13)</code>	14, 10, 8, 1
<code>codes.random_linear_code(GF(3),200,22)</code>	5, 4, 1
<code>codes.random_linear_code(GF(2),78,11)</code>	14, 9, 1
<code>codes.random_linear_code(GF(2),500,100)</code>	23, 21, 20, 1
<code>codes.random_linear_code(GF(5),500,100)</code>	9, 7, 1
<code>codes.random_linear_code(GF(5),300,60)</code>	7, 6, 1
<code>codes.random_linear_code(GF(11),123,13)</code>	1
<code>codes.random_linear_code(GF(13),129,13)</code>	1

**Comparaison du temps d'exécution entre minimum\_distance, Brouwer et Zimmerman:**

<i>Code</i>	<i>minimum_distance</i>	<i>Brouwer</i>	<i>Zimmerman</i>
<code>codes.random_linear_code(GF(13),58,6)</code>	2 s	1min 8s	47.1 s
<code>codes.random_linear_code(GF(25),28,5)</code>	3.08 s	8.11 s	5.69 s
<code>codes.random_linear_code(GF(5),44,15)</code>	time > 5 min		2min 1s
<code>codes.random_linear_code(GF(3), 50, 13)</code>	1.04 s	2.77 s	1.12 s
<code>codes.random_linear_code(GF(3), 73, 15)</code>	8.83 s	58.4 s	20.8 s

codes.random_linear_code(GF(2),111,33)	4min 23s	1min 14s	1min 16s
codes.random_linear_code(GF(4),73,15)	9min 36s		10min 22s
codes.random_linear_code(GF(4), 73, 11)	2.11 s	39.8 s	36.1 s
codes.random_linear_code(GF(3), 49, 17)	1min 10s	15.7 s	3.15 s
codes.random_linear_code(GF(9), 44, 9)	2min 22s	5min 58s	53.8 s
codes.random_linear_code(GF(4), 51, 16)	time > 7 min	1min 48s	1min 47s
codes.random_linear_code(GF(5), 55, 12)	1min 50s	2min 13s	2min 14s
codes.random_linear_code(GF(3), 80, 17)	1min 23s	4min 59s	3min 18s
codes.random_linear_code(GF(2), 135, 32)	2min 34s	1min 50s	1min 51s
codes.random_linear_code(GF(9), 47, 10)	22min 4s	47min 42s	12min 31s
codes.random_linear_code(GF(2), 128, 32)	2min 17s	1min 7s	1min 7s
codes.random_linear_code(GF(7), 31, 8)	2.33 s	8.37 s	1.4 s
codes.random_linear_code(GF(7), 96, 32)	2min 19s	9.56 s	9.82 s
codes.random_linear_code(GF(9), 34, 9)	2min 11s	2min 59s	30.8 s
codes.random_linear_code(GF(7),39,10)	1min 46s	3min 12s	26.4 s
codes.random_linear_code(GF(11),44,8)	1min 14s	7min 53s	5min 23s
codes.random_linear_code(GF(7),49,10)	1min 57s	15min 39s	2min 59s
codes.random_linear_code(GF(7),34,7)	390 ms	3.54 s	960 ms
codes.random_linear_code(GF(7),36,9)	14.6 s	2.22 s	2.24 s
codes.random_linear_code(GF(7),36,11)	12min		2min 20s
codes.random_linear_code(GF(8),39,10)	5min 46s	37min 26s	3min 44s