

Experiment results log.

Planned order of scenarios

Scenarios	Parameters	Week
A1	N	March 23 - March 29
A2	B	March 30 - April 5
A3	Datasets	March 30 - April 5
B1	Chain Strength	April 6 - April 12
B2	Embedding	April 13 - April 19
B3	Shots	April 20 - April 26
B4	Annealing	April 27 - May 3

Actual order of scenarios

Scenarios	Parameters	Week
A1	N	March 23 - March 29
B1	Chain Strength	March 30 - April 5
A2	B	April 6 - April 12
B3	Shots	April 13 - April 19
A2B3	B and Shots	April 20 - April 26
B2	Embedding	April 20 - April 26
B4	Annealing	April 27 - May 3
A3	Datasets	May 4 - May 10

Sidenotes to research about

- Find what is the maximum N value that is supported by dwave

Scenario A1 - N

We started by experimenting several values of N, in order to find the maximum possible value of N that could be solved in a reasonable time by the classical solver.

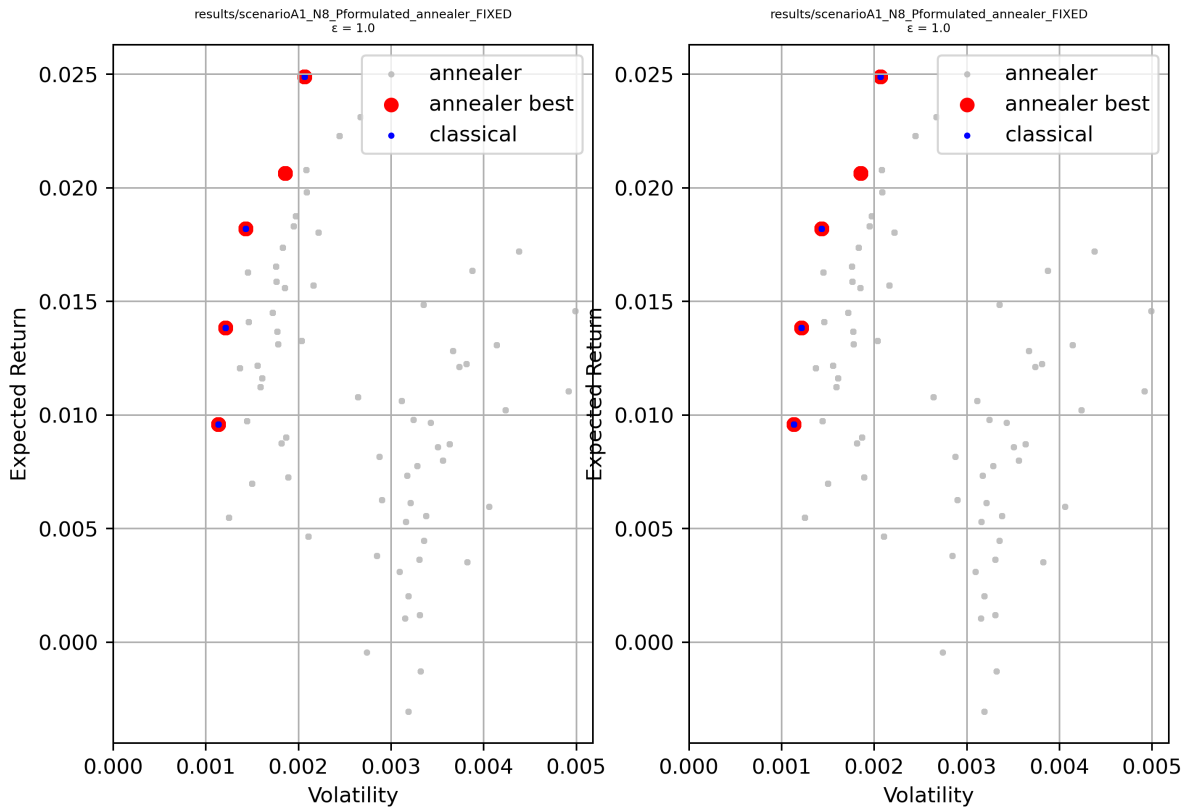
The N values are: 8, 16, 32, and 64. P was calculated as $P = -q * \min_sigma + \max_mu$

For this scenario, we used the "diversified" dataset, 1000 shots per execution, and 5 tries. The q_values are listed in the following table:

N	q values
8	0, 11, 20, 54
16	0, 2, 6, 100, 500
32	0, 0.4, 0.9, 2, 3, 9, 100
64	0, 0.2, 0.4, 0.6, 1.1, 1.3, 1.5, 2, 5, 6, 7, 8, 10, 100, 500

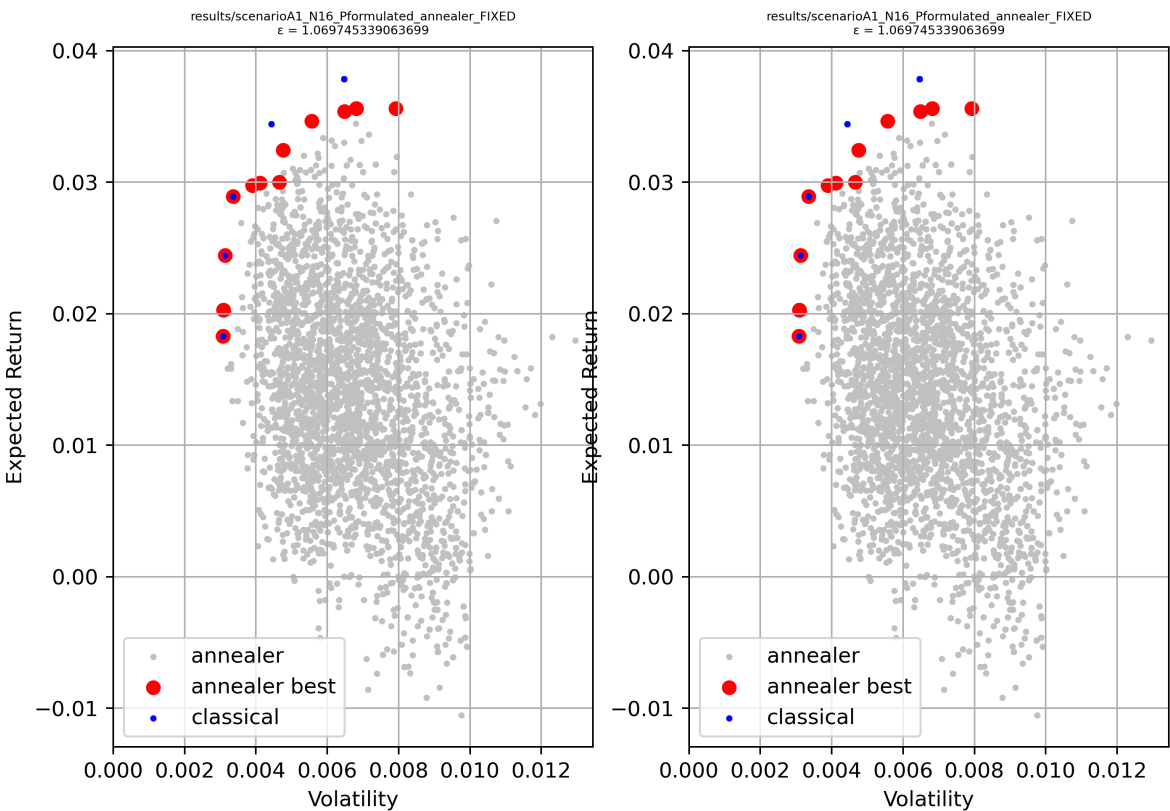
The following images show the first try for each N.

Epsilon Indicator - scenario1Y2021M04D18h23m07s48



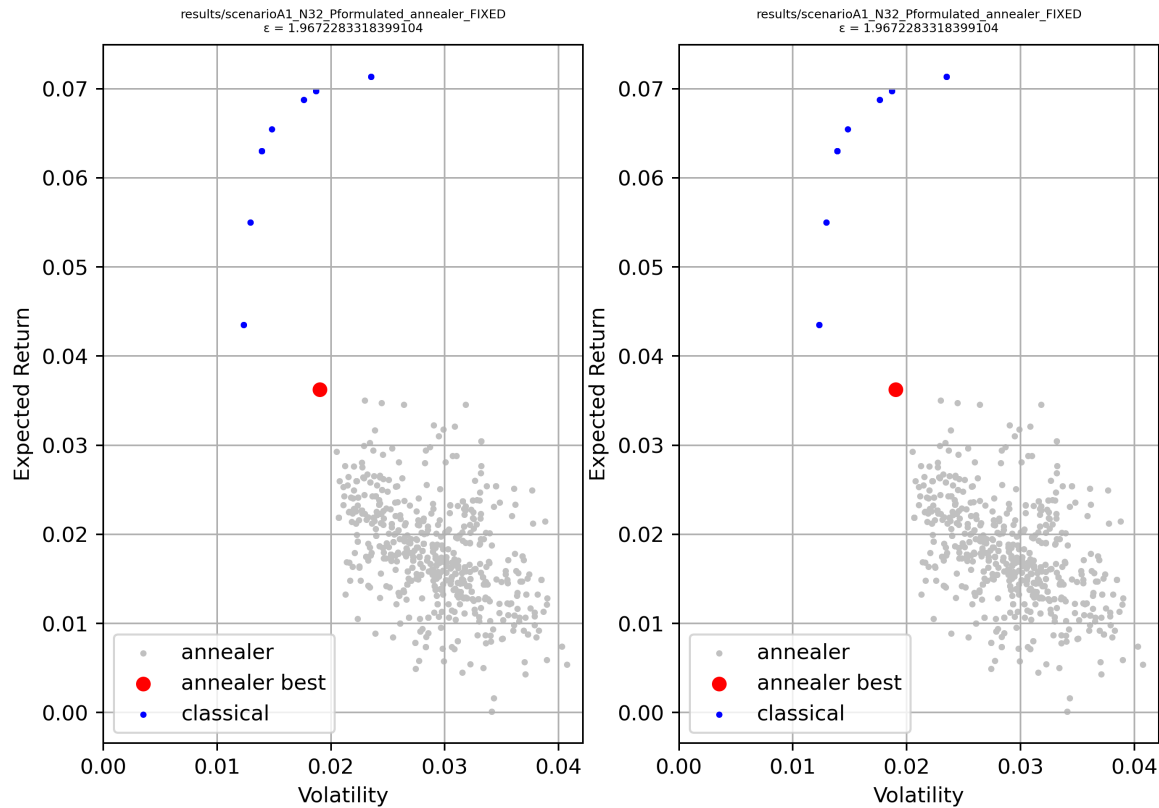
How to interpret: Blue markers are part of the efficient frontier. The epsilon indicator is the minimum factor by which the red set has to be multiplied in the objective so as to weakly dominate Hence, the closer to 1 is the epsilon indicator, the better the red set.

Epsilon Indicator - scenario1Y2021M04D18h23m08s05

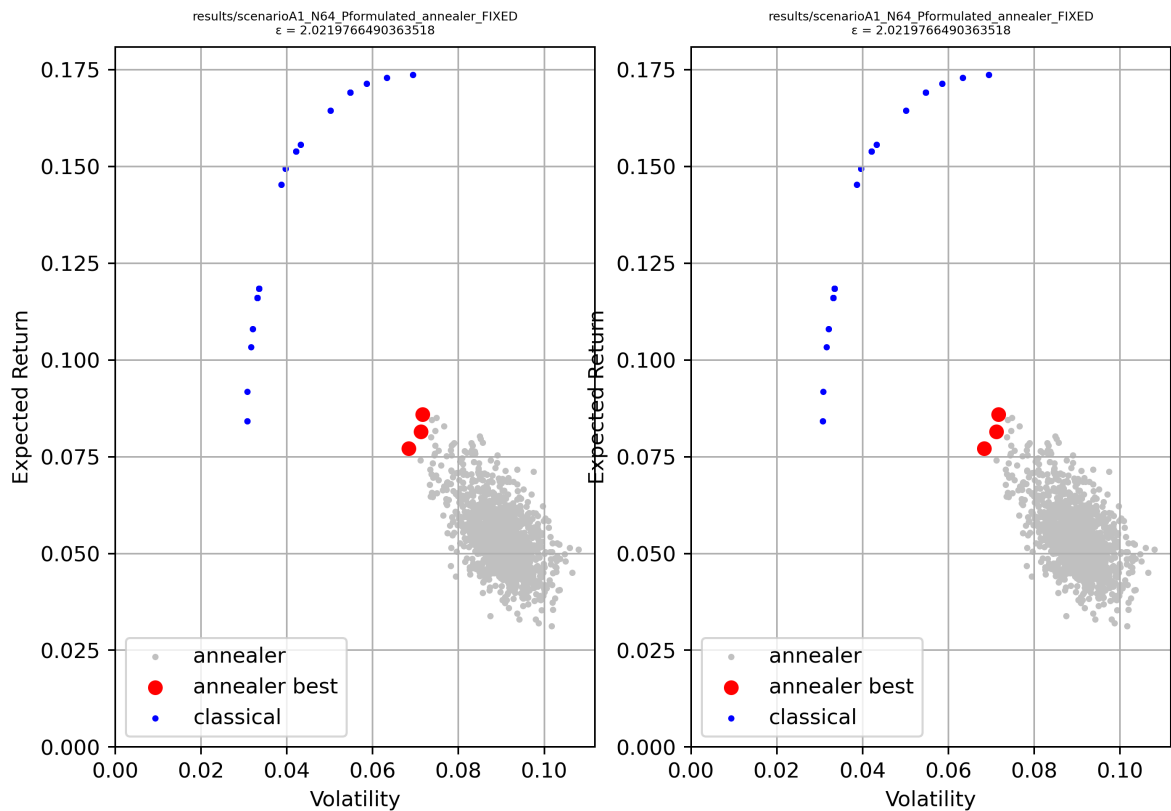


How to interpret: Blue markers are part of the efficient frontier. The epsilon indicator is the minimum factor by which the red set has to be multiplied in the objective so as to weakly dominate Hence, the closer to 1 is the epsilon indicator, the better the red set.

Epsilon Indicator - scenario1Y2021M04D18h23m08s14

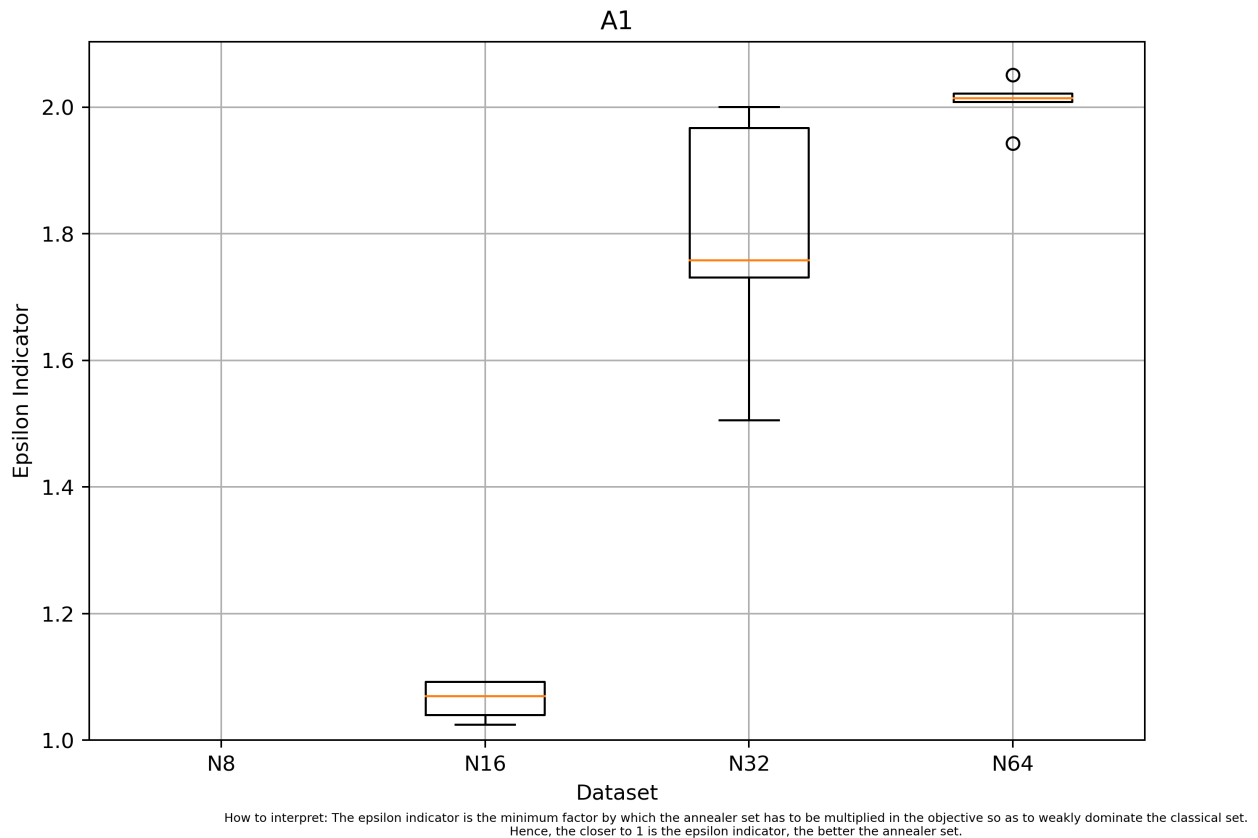


Epsilon Indicator - scenario1Y2021M04D18h23m08s20



Comparing all the different tries, we have:

Boxplots - scenarioA3Y2021M05D10h17m50s29



Hypothesis Testing:

Kruskal-Wallis one-way ANOVA, since we cannot assume the normal distribution of the residuals.

The null hypothesis is that all samples have equal means.

Otherwise, at least two means are different.

We will use a confidence level of 0.05.

For 4 groups with 5 samples each, the critical value for H is 7.377.

Result: `KruskalResult(statistic=17.609472880061105, pvalue=0.0005294252543614154)`

The null hypothesis is rejected.

Post hoc analysis (Conover-Iman Test) needs to be done:

Results of Kruskal-Wallis One-way ANOVA:

H statistic: 17.609472880061105

p-value: 0.0005294252543614154

Null hypothesis rejected! Now performing pairwise comparison with Dunn's test

	N8	N16	N32	N64
N8	1.000000e+00	1.891157e-03	3.330204e-07	2.619788e-09
N16	1.891157e-03	1.000000e+00	8.960526e-04	9.861650e-07
N32	3.330204e-07	8.960526e-04	1.000000e+00	8.708493e-03
N64	2.619788e-09	9.861650e-07	8.708493e-03	1.000000e+00

N8 and N16 ARE significantly different!

N8 and N32 ARE significantly different!
 N8 and N64 ARE significantly different!
 N16 and N32 ARE significantly different!
 N16 and N64 ARE significantly different!
 N32 and N64 ARE significantly different!

Key Takeaways:

As expected, the epsilon indicator increases with the **N** value. However, during those executions, dwave's problem inspector warned that the chains were too weak, and that, in the case of **N=64**, all samples had broken chains. Based on this warning, we decided to immediately execute scenario B1, changing the original order of scenarios.

Scenario B1 - Chain Strength

Looking at the fraction of chain breaks in Scenario A1, we know that on average each sample had almost a third (**0.31**) of its chains broken when **N=32**. This fraction increases to over half (**0.54**) when **N=64**! Those values are very high and are another clue that the chain strength needs to be adjusted, especially for those values of **N**.

A good starting value for the chain strength is the maximum absolute value (**maxAbs**) of the QUBO matrix. However, this is not always the most optimal value. We need to test several values based on this initial value. By testing those values, we can find a value near the sweet spot between the probability that the chains are intact and the probability of finding optimal values. Refer to:

https://www.dwavesys.com/sites/default/files/2_Wed_Am_PerfTips.pdf

We have three tables, one for the epsilon indicator, one for the fractions of valid solutions, and one for the average fractions of chain breaks.

Starting with the average fractions of chain breaks (Lower is better):

Chain strength	N8	N16	N32	N64
default value	0.00081	0.01153	0.31350	0.54426
0.125 * maxAbs	0.00397	0.02741	0.31014	0.38301
0.250 * maxAbs	0.00034	0.00106	0.00170	0.00683
0.375 * maxAbs	0.00006	0.00032	0.00111	0.00453
0.500 * maxAbs	0.00006	0.00026	0.00149	0.00475
0.625 * maxAbs	0.00006	0.00031	0.00112	0.00453
0.750 * maxAbs	0.00006	0.00029	0.00130	0.00454
0.875 * maxAbs	0.00006	0.00017	0.00102	0.00461
1.000 * maxAbs	0.00003	0.00034	0.00100	0.00439
1.125 * maxAbs	0.00000	0.00030	0.00119	0.00401

Chain strength	N8	N16	N32	N64
1.250 * maxAbs	0.00000	0.00042	0.00125	0.00419
1.375 * maxAbs	0.00006	0.00028	0.00108	0.00424
1.500 * maxAbs	0.00009	0.00025	0.00201	0.00430

Next, we obtained the following fractions of valid solutions (Higher is better):

Chain strength	N8	N16	N32	N64
default value	0.877	0.688	0.121	0.094
0.125 * maxAbs	0.001	0.002	0.076	0.205
0.250 * maxAbs	0.934	0.622	0.395	0.243
0.375 * maxAbs	0.848	0.543	0.325	0.220
0.500 * maxAbs	0.781	0.485	0.299	0.186
0.625 * maxAbs	0.703	0.444	0.261	0.172
0.750 * maxAbs	0.665	0.388	0.252	0.170
0.875 * maxAbs	0.630	0.406	0.242	0.163
1.000 * maxAbs	0.598	0.366	0.235	0.151
1.125 * maxAbs	0.594	0.370	0.219	0.148
1.250 * maxAbs	0.556	0.342	0.223	0.129
1.375 * maxAbs	0.540	0.330	0.212	0.136
1.500 * maxAbs	0.512	0.310	0.198	0.138

Finally, we obtained the following epsilon indicators (Lower is better):

Chain strength	N8	N16	N32	N64
default value	1,000	1,070	1,967	2,022
0,125 * maxAbs	1,368	18,844	1,767	1,977
0,250 * maxAbs	1,000	1,075	1,178	1,474
0,375 * maxAbs	1,000	1,057	1,203	1,580
0,500 * maxAbs	1,000	1,099	1,331	1,500
0,625 * maxAbs	1,000	1,098	1,269	1,410
0,750 * maxAbs	1,000	1,120	1,429	1,523
0,875 * maxAbs	1,000	1,123	1,430	1,587
1,000 * maxAbs	1,000	1,119	1,250	1,526

Chain strength	N8	N16	N32	N64
1,125 * maxAbs	1,000	1,099	1,142	1,539
1,250 * maxAbs	1,000	1,092	1,355	1,610
1,375 * maxAbs	1,000	1,110	1,352	1,465
1,500 * maxAbs	1,000	1,101	1,345	1,423

To validate such results, this scenario has been repeated for N=16, N=32, and N=64.

Chain strength	N16	N32	N64
default value	1,092	1,739	1,981
0,125 * maxAbs	2,314	1,813	2,185
0,250 * maxAbs	1,098	1,256	1,550
0,375 * maxAbs	1,109	1,336	1,492
0,500 * maxAbs	1,114	1,257	1,502
0,625 * maxAbs	1,110	1,322	1,503
0,750 * maxAbs	1,077	1,299	1,516
0,875 * maxAbs	1,120	1,307	1,489
1,000 * maxAbs	1,141	1,350	1,485
1,125 * maxAbs	1,114	1,327	1,430
1,250 * maxAbs	1,101	1,266	1,549
1,375 * maxAbs	1,169	1,198	1,508
1,500 * maxAbs	1,126	1,325	1,597

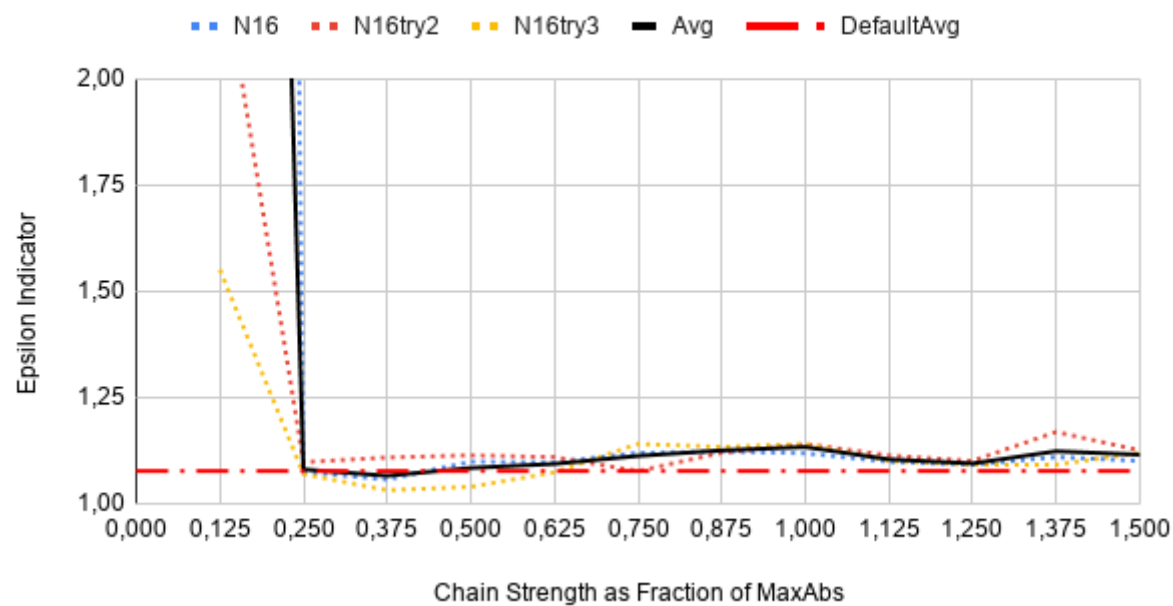
And one more time:

Chain strength	N16	N32	N64
default value	1,070	1,728	1,988
0,125 * maxAbs	1,551	1,760	1,906
0,250 * maxAbs	1,070	1,266	1,462
0,375 * maxAbs	1,032	1,235	1,583
0,500 * maxAbs	1,040	1,325	1,514
0,625 * maxAbs	1,074	1,332	1,551
0,750 * maxAbs	1,141	1,270	1,458
0,875 * maxAbs	1,134	1,229	1,515

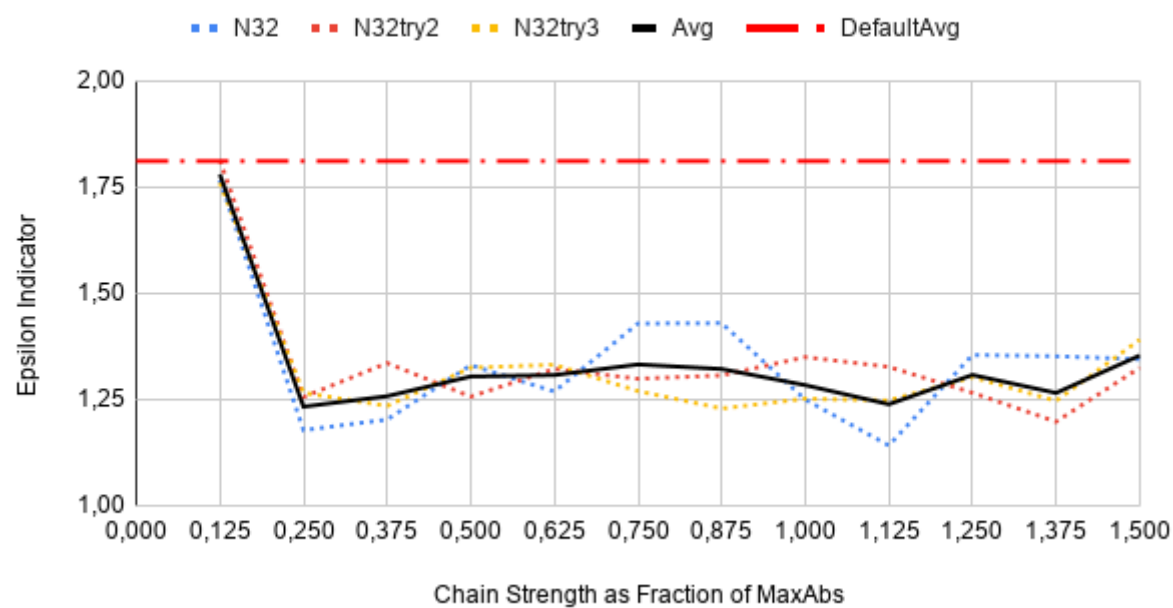
Chain strength	N16	N32	N64
1,000 * maxAbs	1,141	1,252	1,536
1,125 * maxAbs	1,101	1,248	1,547
1,250 * maxAbs	1,092	1,303	1,523
1,375 * maxAbs	1,092	1,247	1,560
1,500 * maxAbs	1,120	1,391	1,519
5,000 * maxAbs	1.177	1,297	1,627

The results are summarized in the following charts.

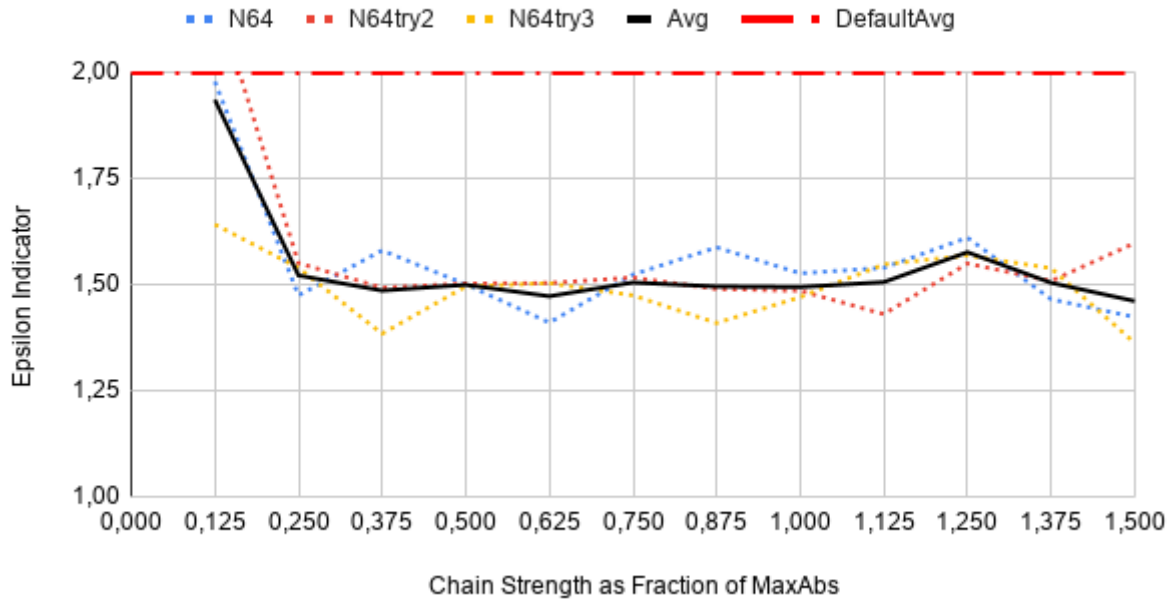
N = 16



N = 32



N = 64



Key Takeaways:

Looking at the results, we notice that the impact of any change to the chain strength is higher for higher values of N .

It also becomes clear that, especially for higher values of N , the default chain strength is far from being the best value. It seems that for higher values of N , the farther is the default chain strength value from the best value.

Another thing that also becomes clear is that the fractions of chain breaks and valid solutions are not directly synonymous with the quality of the solutions.

For the case of $N=8$, every try gave a perfect score of 1.000.

For the case $N=16$, the epsilon values are so similar that they fall under the margin of variation. Thus we cannot place conclusions based on these results. (Note: in this case, the default strength is always the best!)

There is an exception for both cases of $N=8$ and $N=16$. When $\text{chain_strength} = 0.125 * \text{maxAbs}$ there is a high fraction of chain breaks and almost no samples are valid solutions. Thus, for this value of chain strength, the results are very bad.

This behavior is also noticeable for $N=32$ and $N=64$, that present a relatively high epsilon indicator with this chain strength.

It seems that, after this very weak chain strength, the following values of chain strength rapidly attain the lowest epsilon indicators registered, with a very slow climb afterwards.

In the end, the results suggest that it is okay to choose any value that is part of the slow climb. However, from theory, we know that we should avoid any value over $1.000 * \text{maxAbs}$, since it scales down the problem.

Therefore, for all N values, a safe range seems to be between $0.250 * \text{maxAbs}$ and $1.000 * \text{maxAbs}$.

Based on those findings, the case $N=8$ will not be tested in the remaining scenarios, since the annealer already achieved optimality.

Scenario A2 - B OLD

For this scenario, we will be looking at how different budgets affect the performance of the annealer. Therefore, different fractions of B are going to be tested: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9.

I will be using $\text{chain_strength} = 1.000 * \text{maxAbs}$, for the reasons explained in the previous scenario.

Reminder: the fraction used in previous scenarios was $B=0.5$!

Obviously, for each value of B , we first need to solve it classically. Then, from the results, we get the sequences of q_values to be used in the annealer.

N	q values	Budget fraction
16	0, 20, 500	0.1 (1)
32	0, 7, 20, 40	0.1 (3)
64	0, 0.6, 2, 4, 6, 8, 20, 40, 80, 500	0.1 (6)
16	0, 8, 10, 40	0.2 (3)
32	0, 5, 8, 20, 30, 80	0.2 (6)
64	0, 0.3, 0.8, 2, 4, 5, 7, 9, 20, 30, 500	0.2 (12)
16	0, 2, 6, 20, 60	0.3 (4)
32	0, 3, 4, 10, 20, 50	0.3 (9)
64	0, 0.2, 2, 3, 4, 5, 7, 9, 20, 30, 100	0.3 (19)
16	0, 2, 5, 10, 30	0.4 (6)
32	0, 0.2, 0.9, 2, 4, 20, 30, 70, 500	0.4 (12)
64	0, 0.3, 0.6, 1, 2, 3, 4, 6, 8, 20, 30, 90	0.4 (25)
16	0, 2, 6, 100, 500	0.5 (8)
32	0, 0.4, 0.9, 2, 3, 9, 100	0.5 (16)
64	0, 0.2, 0.4, 0.6, 1.1, 1.3, 1.5, 2, 5, 6, 7, 8, 10, 100, 500	0.5 (32)
16	0, 0.1, 0.8, 3, 20, 30	0.6 (9)
32	0, 0.1, 0.5, 1, 2, 3, 7, 8, 20, 30	0.6 (19)
64	0, 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 2, 3, 7, 9, 20	0.6 (38)
16	0, 0.7, 20	0.7 (11)
32	0, 0.4, 2	0.7 (22)
64	0, 0.1, 0.2, 0.3, 0.7, 1, 2, 3, 4, 6, 20	0.7 (44)

N	q values	Budget fraction
16	0, 4	0.8 (12)
32	0, 0.8, 7, 9	0.8 (25)
64	0, 0.1, 0.2, 0.4, 0.5, 0.6, 1, 2, 3, 6, 20	0.8 (51)
16	0, 50	0.9 (14)
32	0, 0.8, 3	0.9 (28)
64	0, 0.6, 1, 2, 5, 500	0.9 (57)

Question: Is it bad to have different number of samples between cases?

With those results, we obtained the following epsilon indicators:

Budget fraction	N16 (AvgChainBreak)	N32 (AvgChainBreak)	N64 (AvgChainBreak)
0,1	1,000 (0,00406)	1,142 (0,00979)	1,846 (0,01572)
0,2	1,000 (0,00048)	1,334 (0,00151)	2,929 (0,00493)
0,3	1,026 (0,00044)	1,373 (0,00152)	1,750 (0,00473)
0,4	1,113 (0,00024)	1,281 (0,00134)	1,640 (0,00452)
0,5	1,075 (0,00031)	1,311 (0,00127)	1,513 (0,00484)
0,6	1,146 (0,00033)	1,293 (0,00141)	1,441 (0,00470)
0,7	1,162 (0,00038)	1,421 (0,00108)	1,907 (0,00464)
0,8	1,005 (0,00031)	1,408 (0,00144)	inf (0,00462)
0,9	1,103 (0,00034)	inf (0,00129)	inf (0,00447)

Gráficos com os resultados deste cenário estão no cenário seguinte

Key Takeaways:

The first thing I notice is that there is a high chain break fraction when the budget is $B=0.1$. Afterwards, it attains a consistently low fraction, with small variation.

For $N=32$ and $N=64$, as expected from theory, the epsilon indicator is lower when the budget is or is close to $B=0.5$, since this value has the highest number of admissible solutions.

Behavior for all N values is hard to grasp. Nonetheless, budget fraction is a parameter that is particular to each practitioner.

Scenario B3 - Shots **OLD**

The previous scenario, A2, made us wonder about the number of samples. That is, there is a possibility that the cases where B is farthest from $B=0.5$ have worse performance because of having less values of q and thus less samples taken.

Therefore, we pose a question: Is it better to increase the number of shots per value of q or to add more values of q to be executed?

Since the results so far seem to have a good coverage of the efficient frontier, but still far from it, we believe that the issue is related to the number of samples per value of q . Hence, we are going to repeat the previous scenario with a new methodology to define the number of samples per value of q . This methodology is called **Allocated**.

Initially, each value of q had 1000 shots, i.e., 1000 samples taken. This time, each case will have a total allocated number of shots for every value of q . For example, if we have a case with three values of q and another case with five values of q , then, with a total allocation of 5000 shots per case, then the first case will have 1666 shots per value, while the second case will have 1000 shots per value.

Based on this methodology, we will start with a total allocation of 15000 shots, such that each of the 15 values of q from case $B=0.5$ have 1000 shots.

N	q values	Budget fraction	Shots per value of q
16	0, 20, 500	0.1 (1)	5000
32	0, 7, 20, 40	0.1 (3)	3750
64	0, 0.6, 2, 4, 6, 8, 20, 40, 80, 500	0.1 (6)	1500
16	0, 8, 10, 40	0.2 (3)	3750
32	0, 5, 8, 20, 30, 80	0.2 (6)	2500
64	0, 0.3, 0.8, 2, 4, 5, 7, 9, 20, 30, 500	0.2 (12)	1363
16	0, 2, 6, 20, 60	0.3 (4)	3000
32	0, 3, 4, 10, 20, 50	0.3 (9)	2500
64	0, 0.2, 2, 3, 4, 5, 7, 9, 20, 30, 100	0.3 (19)	1363
16	0, 2, 5, 10, 30	0.4 (6)	3000
32	0, 0.2, 0.9, 2, 4, 20, 30, 70, 500	0.4 (12)	1666
64	0, 0.3, 0.6, 1, 2, 3, 4, 6, 8, 20, 30, 90	0.4 (25)	1250
16	0, 2, 6, 100, 500	0.5 (8)	3000
32	0, 0.4, 0.9, 2, 3, 9, 100	0.5 (16)	2142
64	0, 0.2, 0.4, 0.6, 1.1, 1.3, 1.5, 2, 5, 6, 7, 8, 10, 100, 500	0.5 (32)	1000
16	0, 0.1, 0.8, 3, 20, 30	0.6 (9)	2500
32	0, 0.1, 0.5, 1, 2, 3, 7, 8, 20, 30	0.6 (19)	1500
64	0, 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 2, 3, 7, 9, 20	0.6 (38)	1250
16	0, 0.7, 20	0.7 (11)	5000
32	0, 0.4, 2	0.7 (22)	5000

N	q values	Budget fraction	Shots per value of q
64	0, 0.1, 0.2, 0.3, 0.7, 1, 2, 3, 4, 6, 20	0.7 (44)	1363
16	0, 4	0.8 (12)	7500
32	0, 0.8, 7, 9	0.8 (25)	3750
64	0, 0.1, 0.2, 0.4, 0.5, 0.6, 1, 2, 3, 6, 20	0.8 (51)	1363
16	0, 50	0.9 (14)	7500
32	0, 0.8, 3	0.9 (28)	5000
64	0, 0.6, 1, 2, 5, 500	0.9 (57)	2500

We obtained the following epsilon indicators:

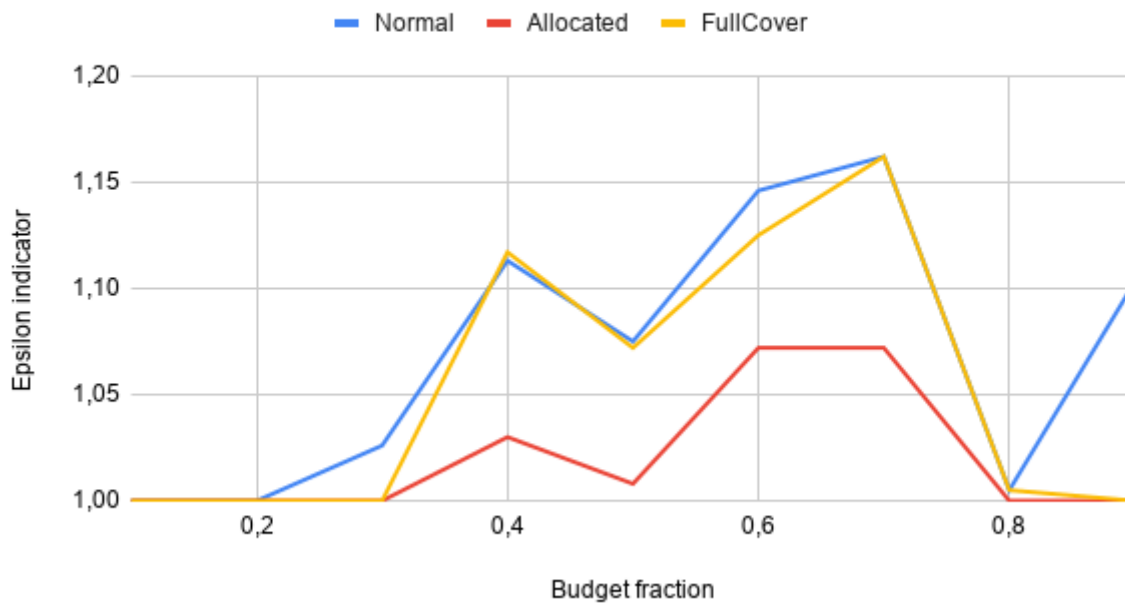
Budget fraction	N16	N32	N64
0,1	1,000	1,069	1,667
0,2	1,000	1,295	2,434
0,3	1,000	1,293	1,651
0,4	1,030	1,277	1,632
0,5	1,008	1,271	1,507
0,6	1,072	1,298	1,521
0,7	1,072	1,333	1,926
0,8	1,000	1,413	inf
0,9	1,000	inf	inf

However, we need to take into account that in real case scenarios, we won't be able to have these carefully chosen values of q. In fact, they were discovered because it was feasible to classically solve these scenarios! For this reason, we introduce another methodology, called **FullCoverage**. This methodology will execute the same values of q for every scenario. The list of values of q is based on guesswork and gained experience with the given scenarios: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 1000. As with **Allocated** methodology, this list is allocated to a total of 15000 samples (500 per value of q).

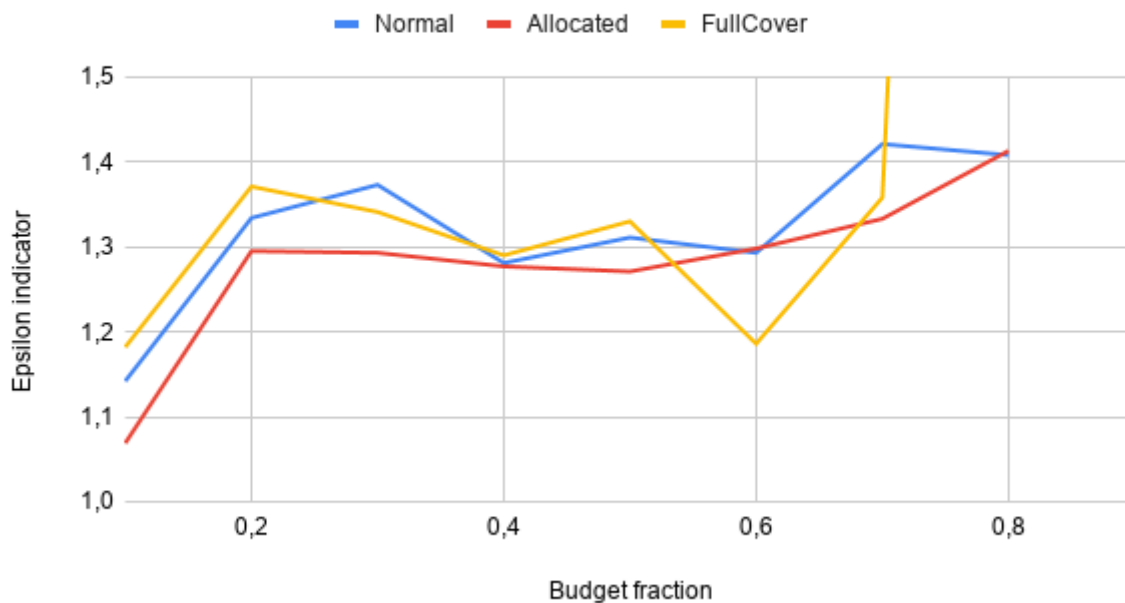
Budget fraction	N16	N32	N64
0,1	1,000	1,182	5,000
0,2	1,000	1,371	4,625
0,3	1,000	1,341	1,739
0,4	1,117	1,290	1,625
0,5	1,072	1,330	1,530

Budget fraction	N16	N32	N64
0,6	1,125	1,186	1,389
0,7	1,162	1,358	1,626
0,8	1,005	4,442	inf
0,9	1,000	1,374	inf

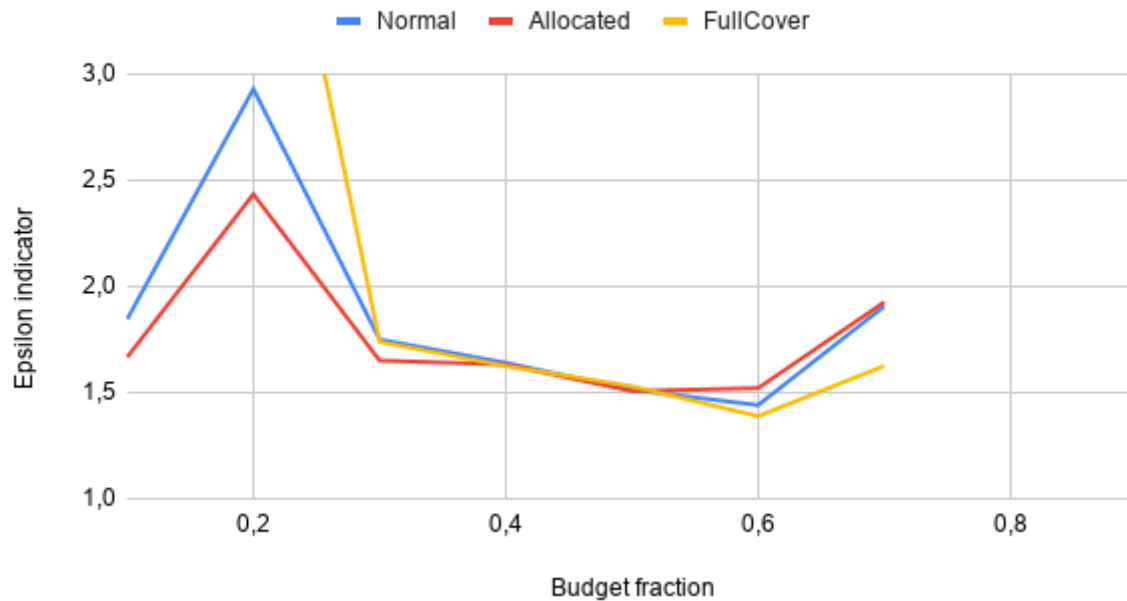
N = 16



N = 32



N = 64



Key Takeaways:

Compared to the previous methodology, called **Simple**, the **Allocated** methodology brings improvements in almost every case. This is expected, since all the cases had their number of samples increased, minus the case **N=64 B=0.5**, which keeps the same number of samples (and also has the same performance in both methodologies).

When looking at the more "realistic" **FullCover** methodology, the results are not the best, but don't fall shortly compared to **Allocated**.

For the next scenarios, we are going to use the **Allocated** methodology, as well as **B=0.5**.

Scenario A2 and B3 - B and Shots

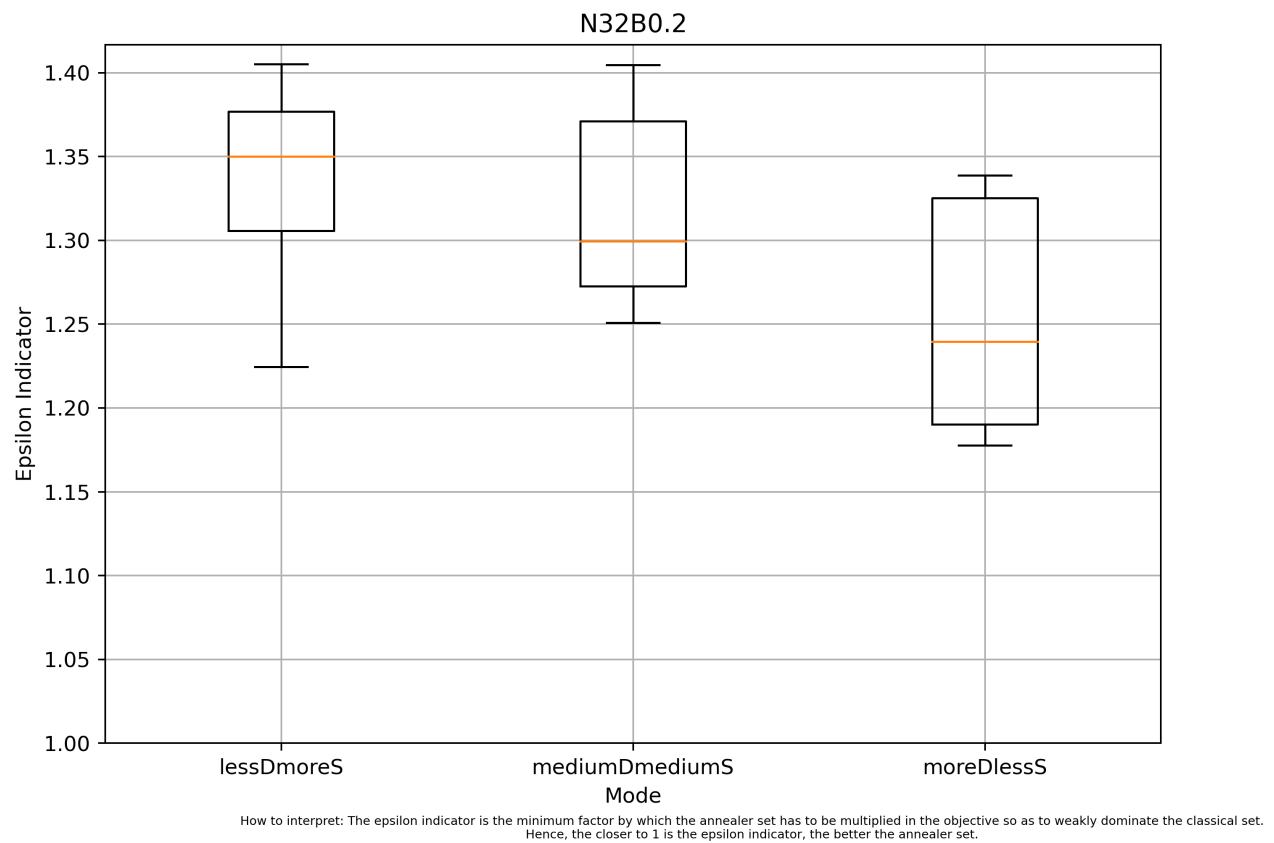
2 factors: B and Shots

"B" factor has three levels: Small Budget, Medium Budget and Large Budget (fractions 0.2, 0.5, and 0.8, respectively).

"Shots" factor has three levels: Less directions and More shots per direction, Medium directions and Medium shots per direction, More directions and Less shots per direction (codenamed **lessDmoreS**, **mediumDmediumS**, and **moreDlessS**, respectively).

Starting with **N=32**:

Boxplots - scenario1Y2021M04D27h19m08s33



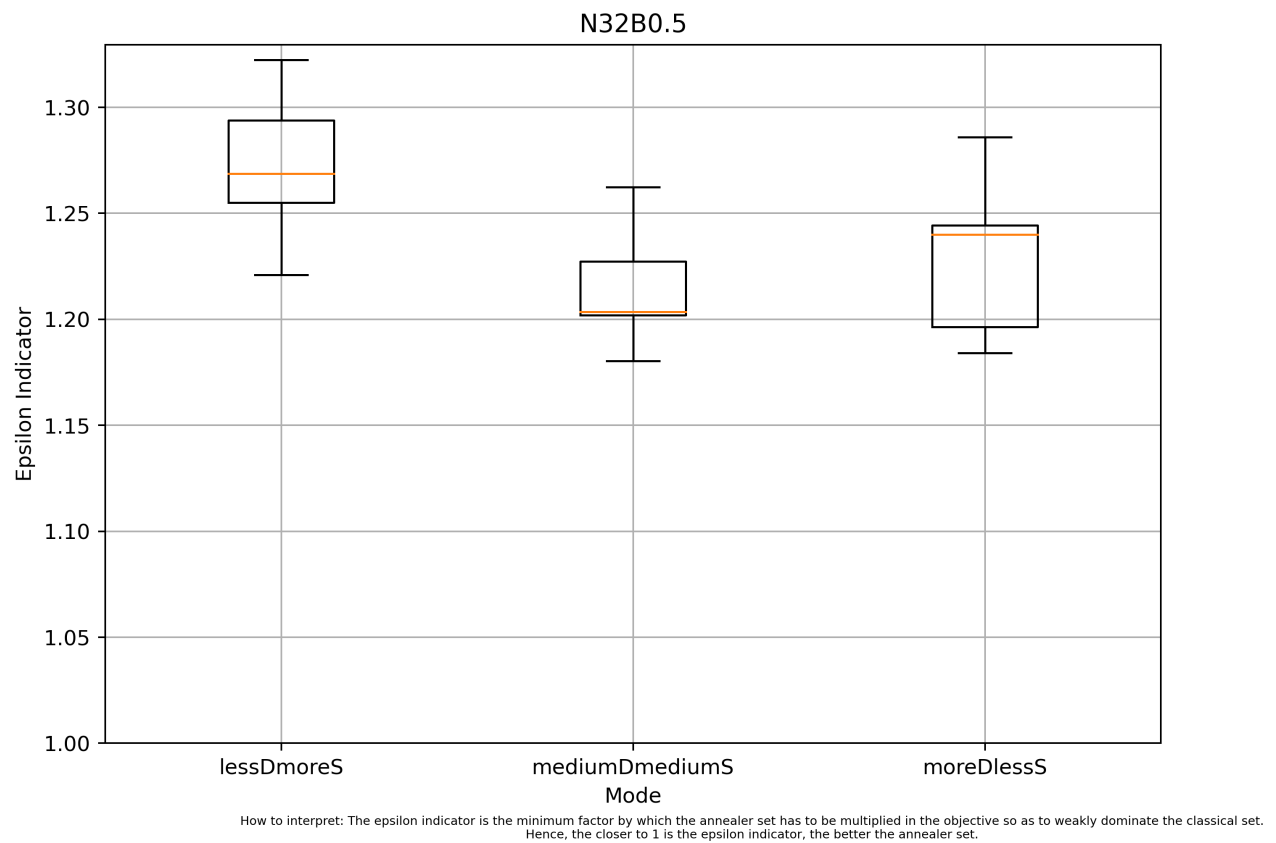
Results of Kruskal-Wallis One-way ANOVA:

H statistic: 3.1200000000000045

p-value: 0.21013607120076422

The null hypothesis was not rejected!

Boxplots - scenario1Y2021M04D27h19m09s04



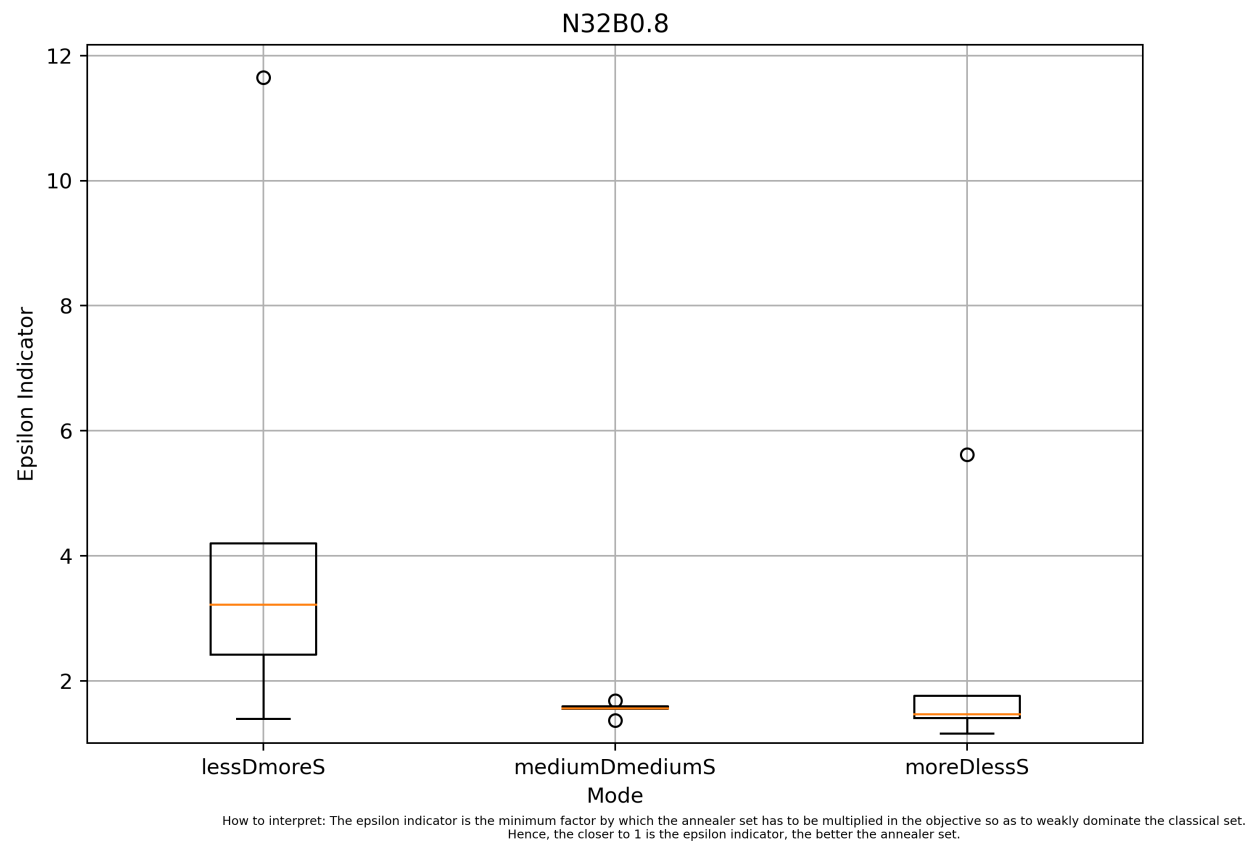
Results of Kruskal-Wallis One-way ANOVA:

H statistic: 4.579999999999998

p-value: 0.1012664618538835

The null hypothesis was not rejected!

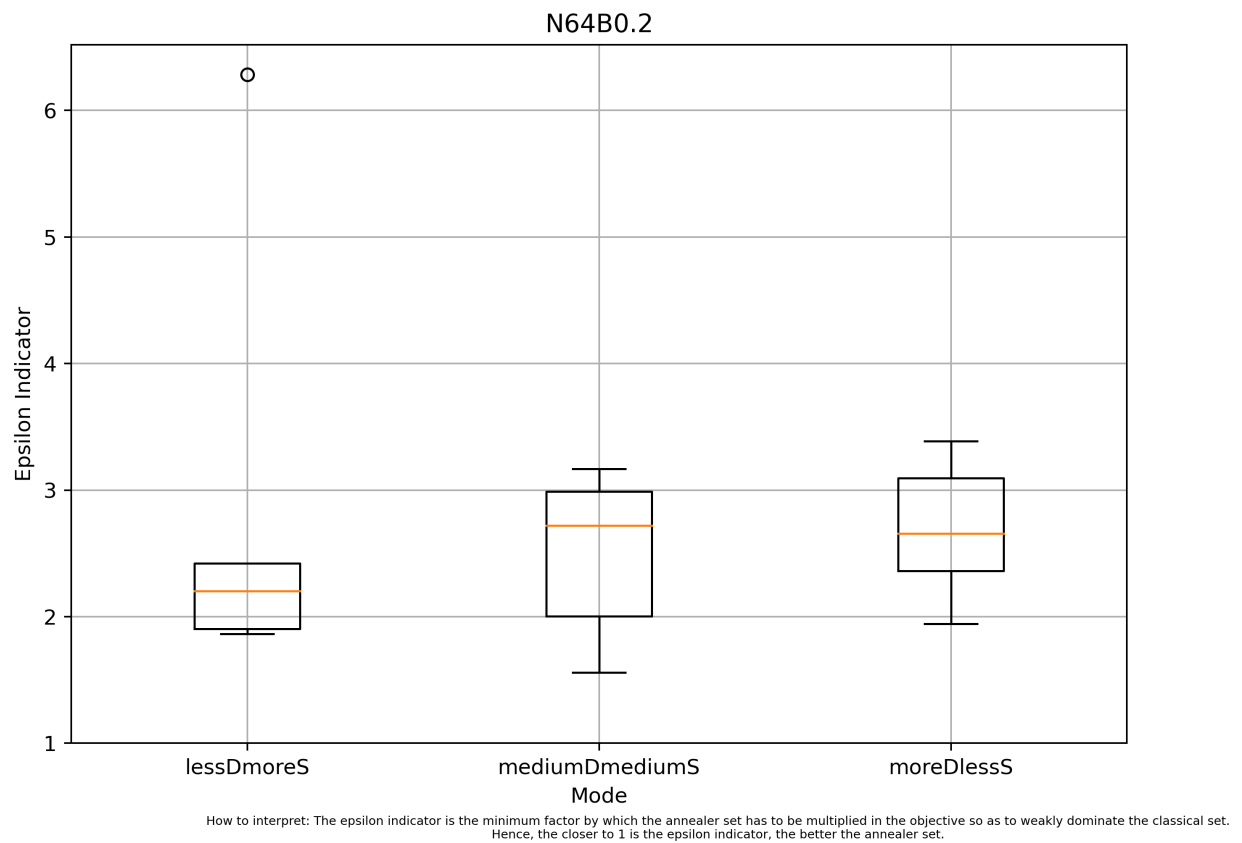
Boxplots - scenario1Y2021M04D27h19m05s21



Results of Kruskal-Wallis One-way ANOVA:
H statistic: 2.9600000000000008
p-value: 0.22763768838381188
The null hypothesis was not rejected!

And for N=64:

Boxplots - scenario1Y2021M04D27h19m04s33



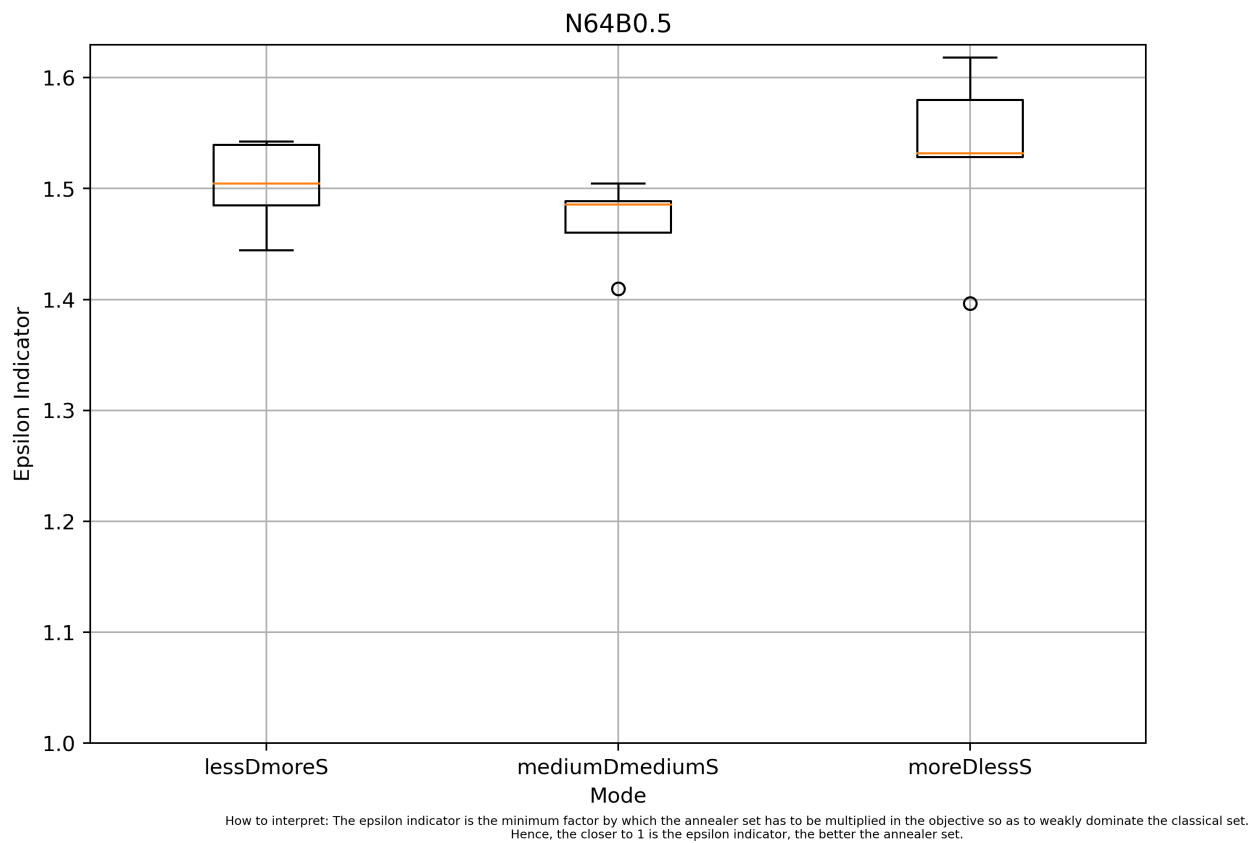
Results of Kruskal-Wallis One-way ANOVA:

H statistic: 0.720000000000006

p-value: 0.697676326071029

The null hypothesis was not rejected!

Boxplots - scenario1Y2021M04D27h19m09s24



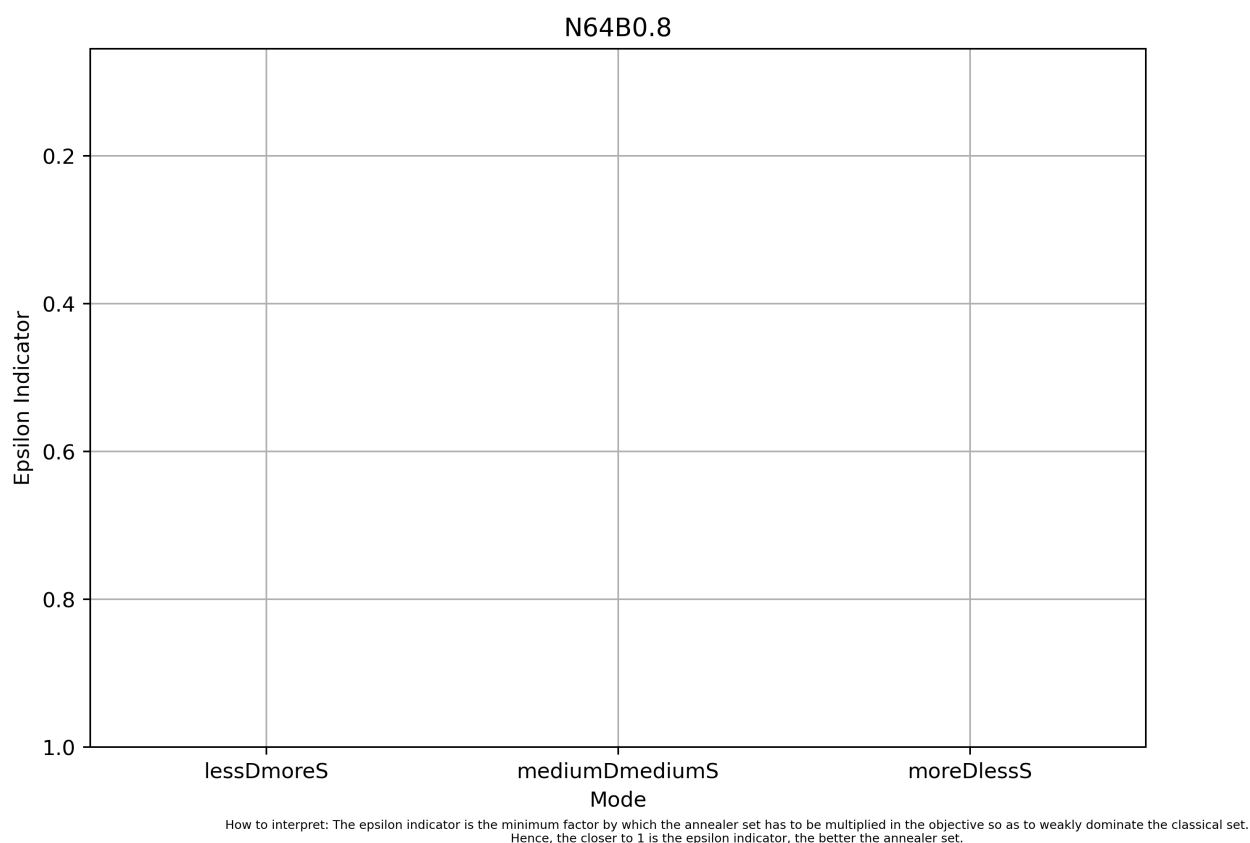
Results of Kruskal-Wallis One-way ANOVA:

H statistic: 2.6600000000000037

p-value: 0.2644772612998236

The null hypothesis was not rejected!

Boxplots - scenario1Y2021M04D27h19m04s58



Results of Kruskal-Wallis One-way ANOVA:

H statistic: 2.0

p-value: 0.36787944117144245

The null hypothesis was not rejected!

Key Takeaways:

No significant differences.

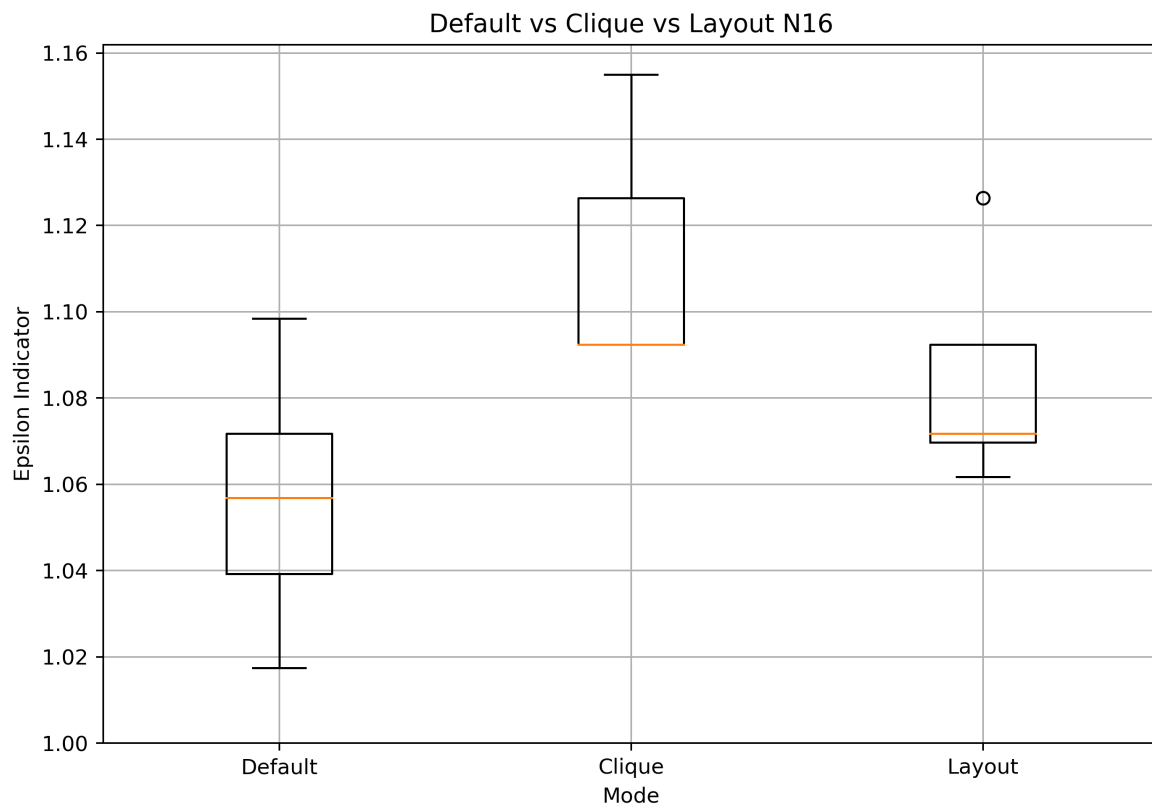
Scenario B2 - Embedding

So far, we used the **general** embedding. D-Wave offers another two embedding options, **clique** and **layout** embeddings. The three options are going to be compared.

Embedding	N16	N32	N64
general try1	1,057	1,165	1,593
general try2	1,039	1,275	1,548
general try3	1,072	1,275	1,568
general try4	1,017	1,290	1,487
general try5	1,098	1,252	1,335
clique try1	1,092	1,316	1,546

Embedding	N16	N32	N64
clique try2	1,092	1,320	1,510
clique try3	1,155	1,381	1,428
clique try4	1,126	1,316	1,577
clique try5	1,092	1,254	1,518
layout try1	1.070	1.250	1.454
layout try2	1.072	1.326	1.389
layout try3	1.062	1.301	1.464
layout try4	1.092	1.248	1.472
layout try5	1.126	1.275	1.459

Boxplots - scenario1Y2021M04D30h16m33s35



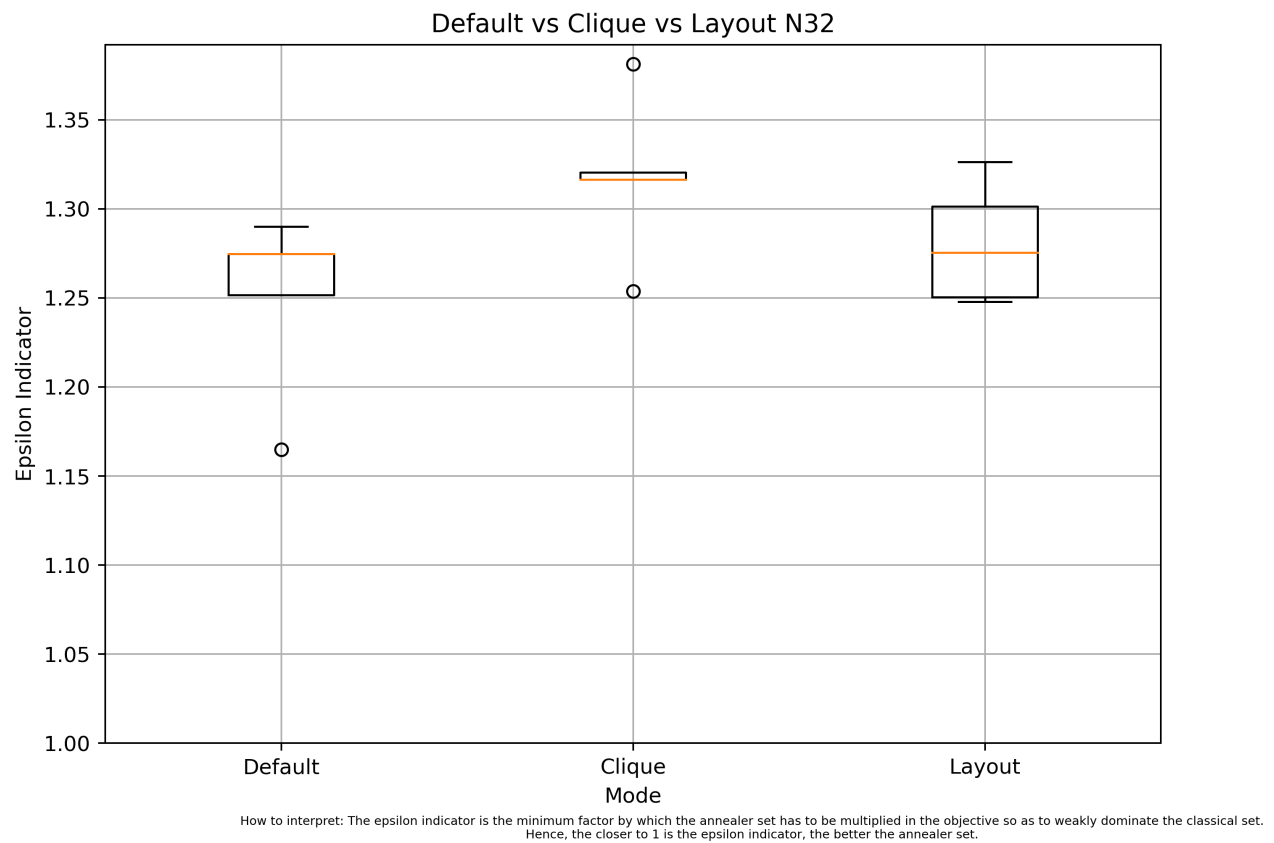
Results of Kruskal-Wallis One-way ANOVA:

H statistic: 5.431386861313866

p-value: 0.06615906037653473

The null hypothesis was not rejected!

Boxplots - scenario1Y2021M04D30h16m33s10



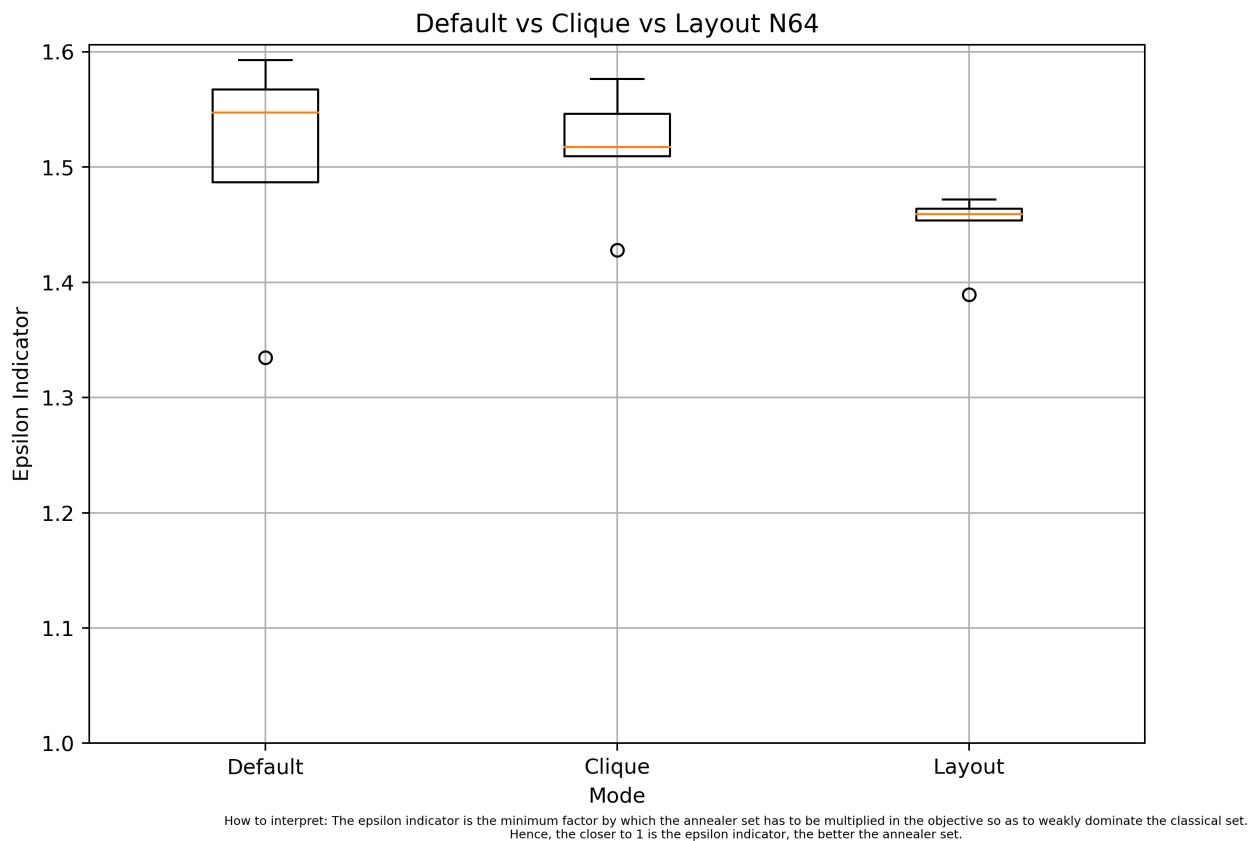
Results of Kruskal-Wallis One-way ANOVA:

H statistic: 4.340000000000003

p-value: 0.11417761691083628

The null hypothesis was not rejected!

Boxplots - scenario1Y2021M04D30h16m32s38



Results of Kruskal-Wallis One-way ANOVA:

H statistic: 3.8600000000000065

p-value: 0.14514819848362326

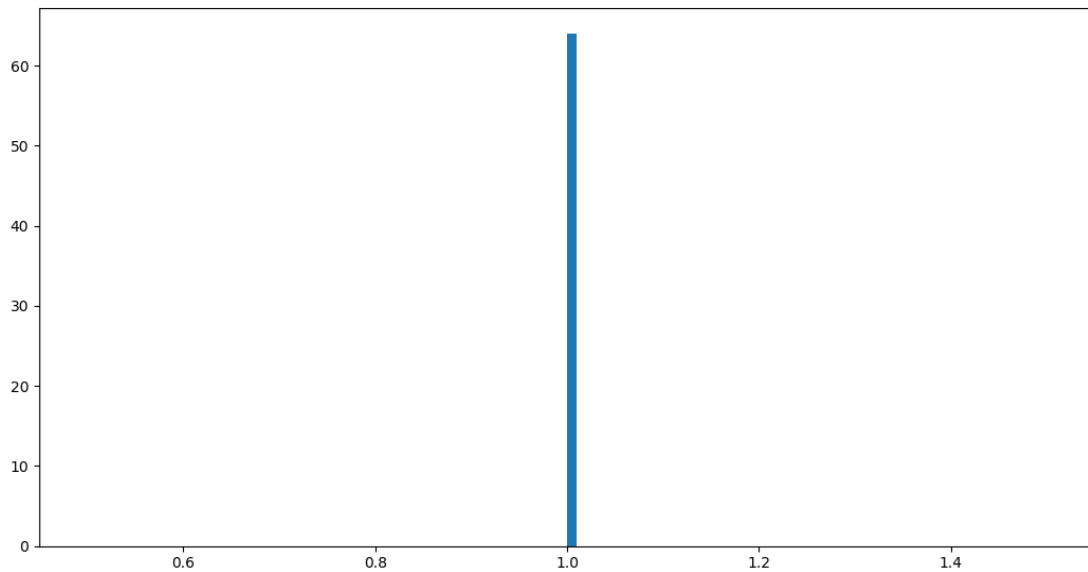
The null hypothesis was not rejected!

Key Takeaways:

No significant difference.

In conversations with Jose Pinilla, a Ph.D. student that authored an implementation of a layout-aware embedding, **layout** embedding is much more suited for *sparse* graphs, which is not the case of the POP. In fact, POP usually generates fully connected graphs. However, Jose Pinilla said "if there are clusters of high connectivity, you'll immediately be rewarded with faster results, or a higher chance of at least finding an embedding". I noticed that, in fact, **layout** embedding was much faster than the other two options.

It is interesting that those faster results were also accompanied by better performance. Again, in conversations with Jose Pinilla, he provided me with some code to plot a histogram that let us confirm that the graph is in fact fully connected.



The graph is fully connected, which means that there are no clusters of high connectivity and no speed boost should be expected. **So, why did it have better performance? This is an interesting question that I pose for further research.**

For the remaining scenarios, we will use the `layout` embedding.

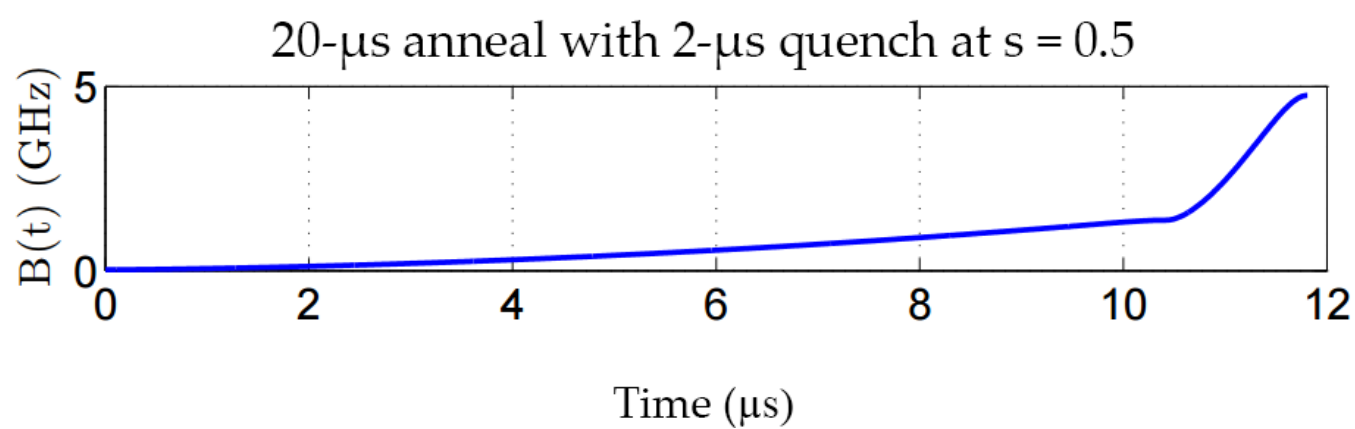
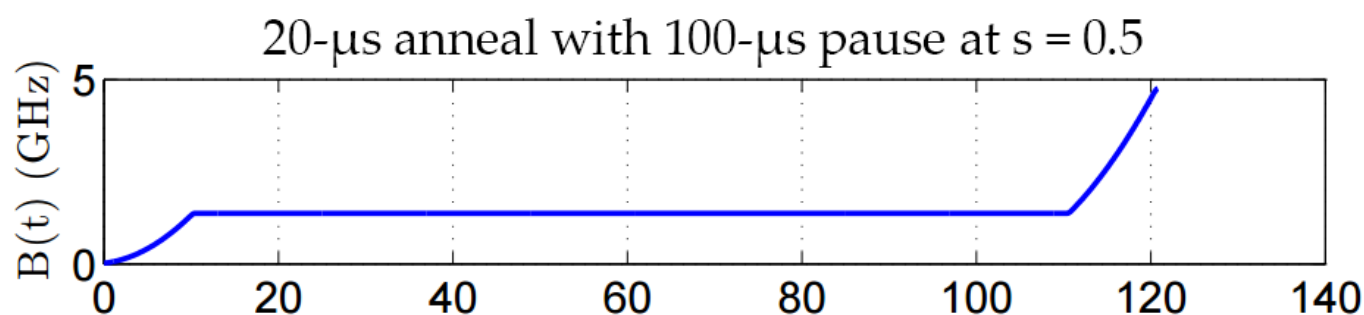
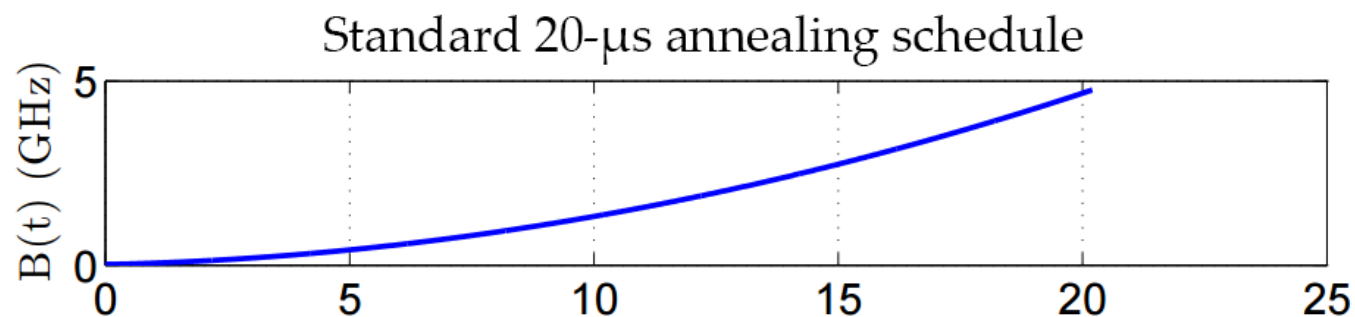
Scenario B4 - Annealing

It is time to study the impact of Annealing, if it has any!

So far, we used the `default` annealing strategy. We will study another three common strategies that may provide significant improvements to the annealer performance.

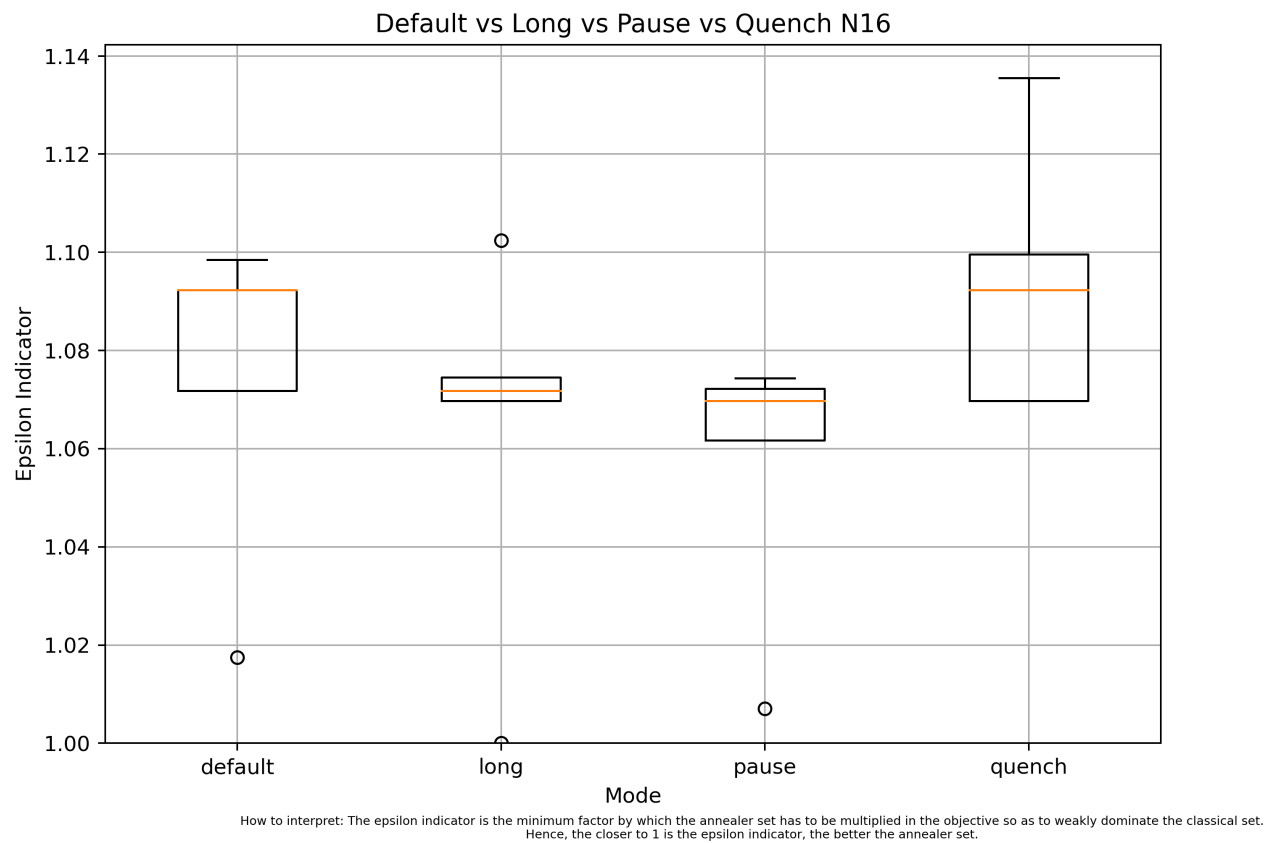
- `default` Standard 20 μ s annealing schedule
- `long` Standard 100 μ s annealing schedule
- `pause` 20 μ s anneal with 100 μ s pause at $s=0.5$
- `quench` 20 μ s anneal with 2 μ s quench at $s=0.5$

`default`, `pause`, and `quench` are illustrated in the following image:



The experiments were run five times:

Boxplots - scenario1Y2021M05D01h02m05s12

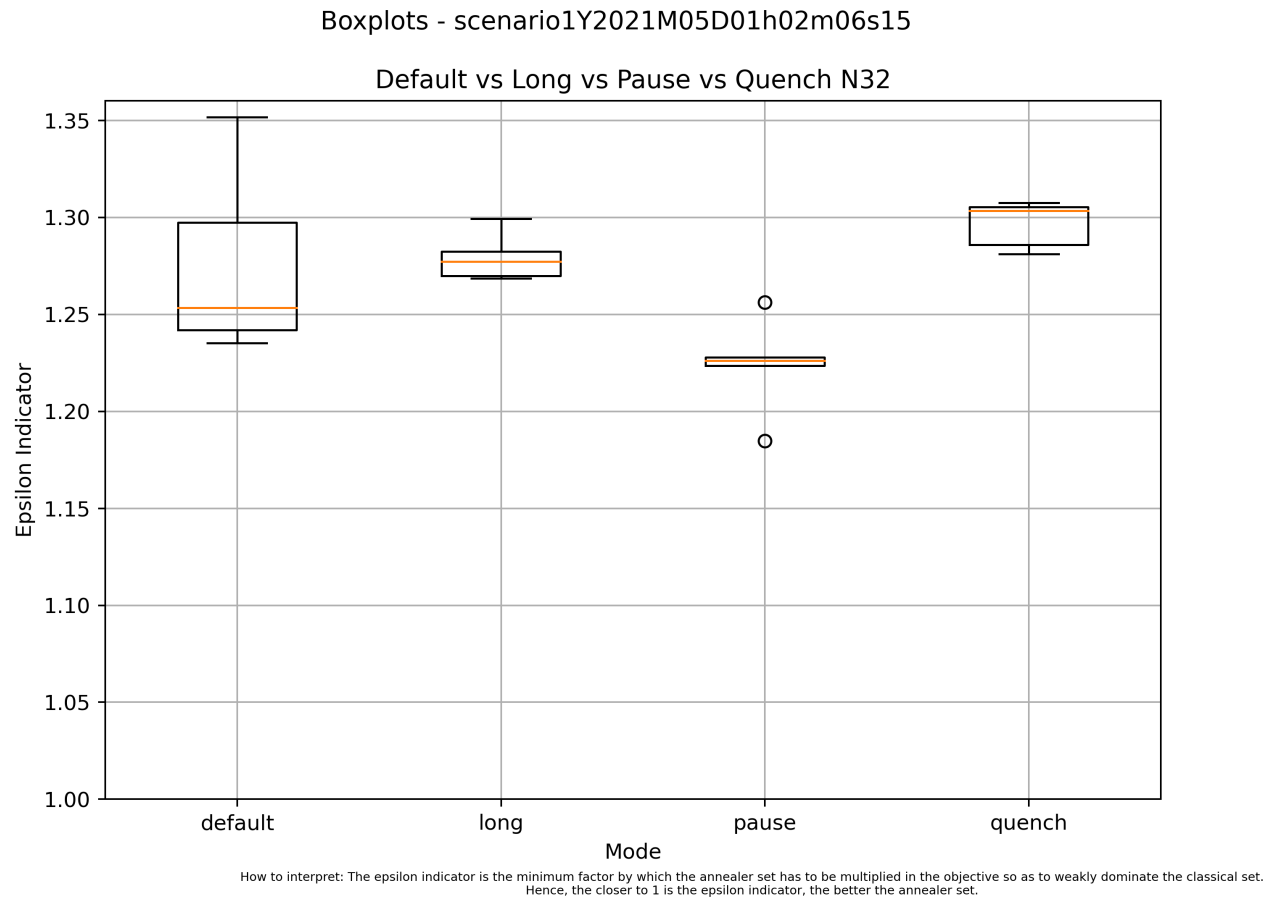


Results of Kruskal-Wallis One-way ANOVA:

H statistic: 2.9657834973504804

p-value: 0.3969308295543481

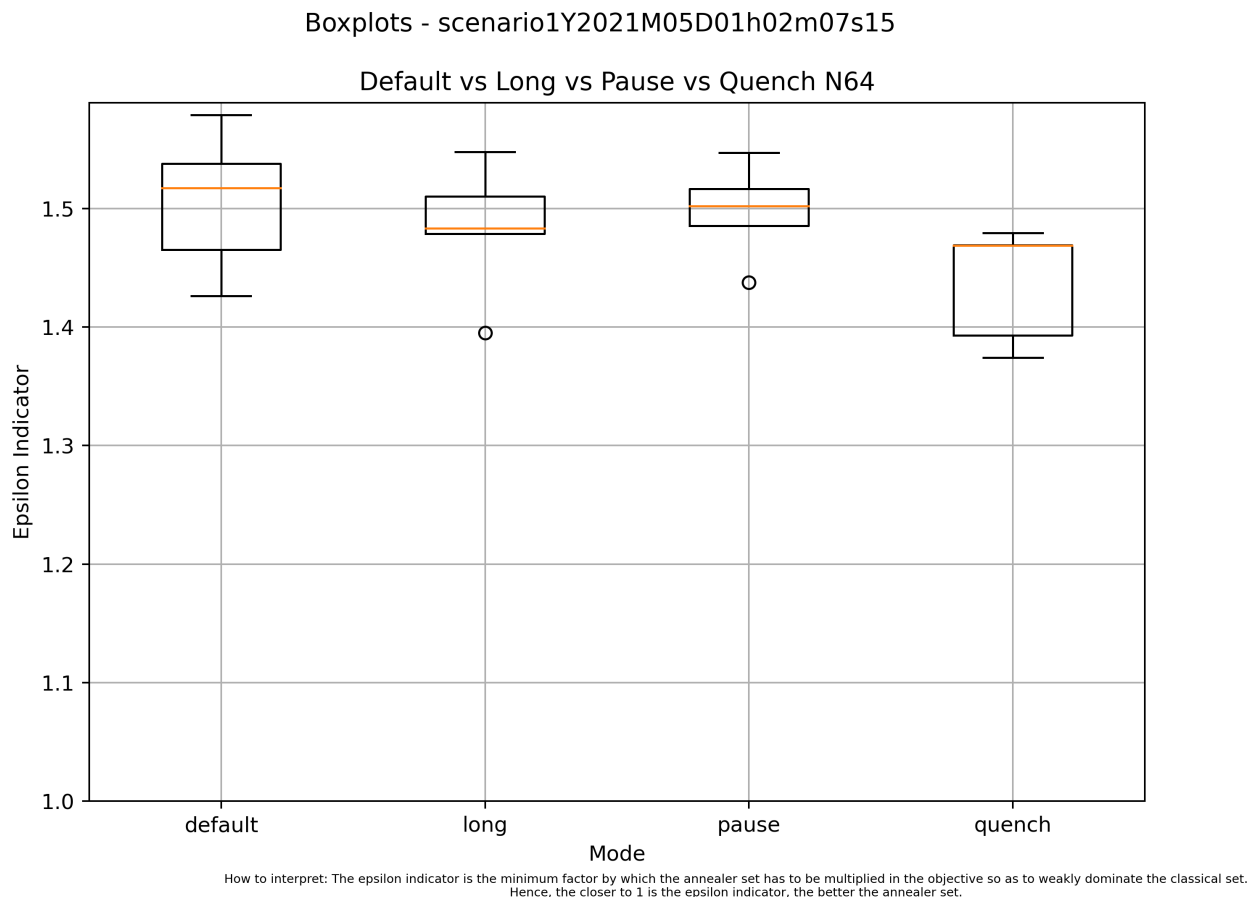
The null hypothesis was not rejected!



Results of Kruskal-Wallis One-way ANOVA:
H statistic: 11.36571428571429
p-value: 0.009904105007103491
Null hypothesis rejected! Now performing pairwise comparison with Dunn's test

	default	long	pause	quench
default	1.000000	1.000000	0.093021	0.318058
long	1.000000	1.000000	0.035443	0.742135
pause	0.093021	0.035443	1.000000	0.001183
quench	0.318058	0.742135	0.001183	1.000000

default and long are NOT!
default and pause are NOT!
default and quench are NOT!
long and pause ARE significantly different!
long and quench are NOT!
pause and quench ARE significantly different!



Results of Kruskal-Wallis One-way ANOVA:

H statistic: 4.760000000000005

p-value: 0.19023861235673986

The null hypothesis was not rejected!

Key Takeaways:

As expected, **long** and **pause** are consistently better than **default**, since they have at least as much anneal time as **default**. This is, however, at the cost of more machine time budget. In fact, **long** achieved a perfect score at **N=16** in one of the runs.

quench is interesting, since it falls short when **N=16** and **N=32**, but outperforms when **N=64**.

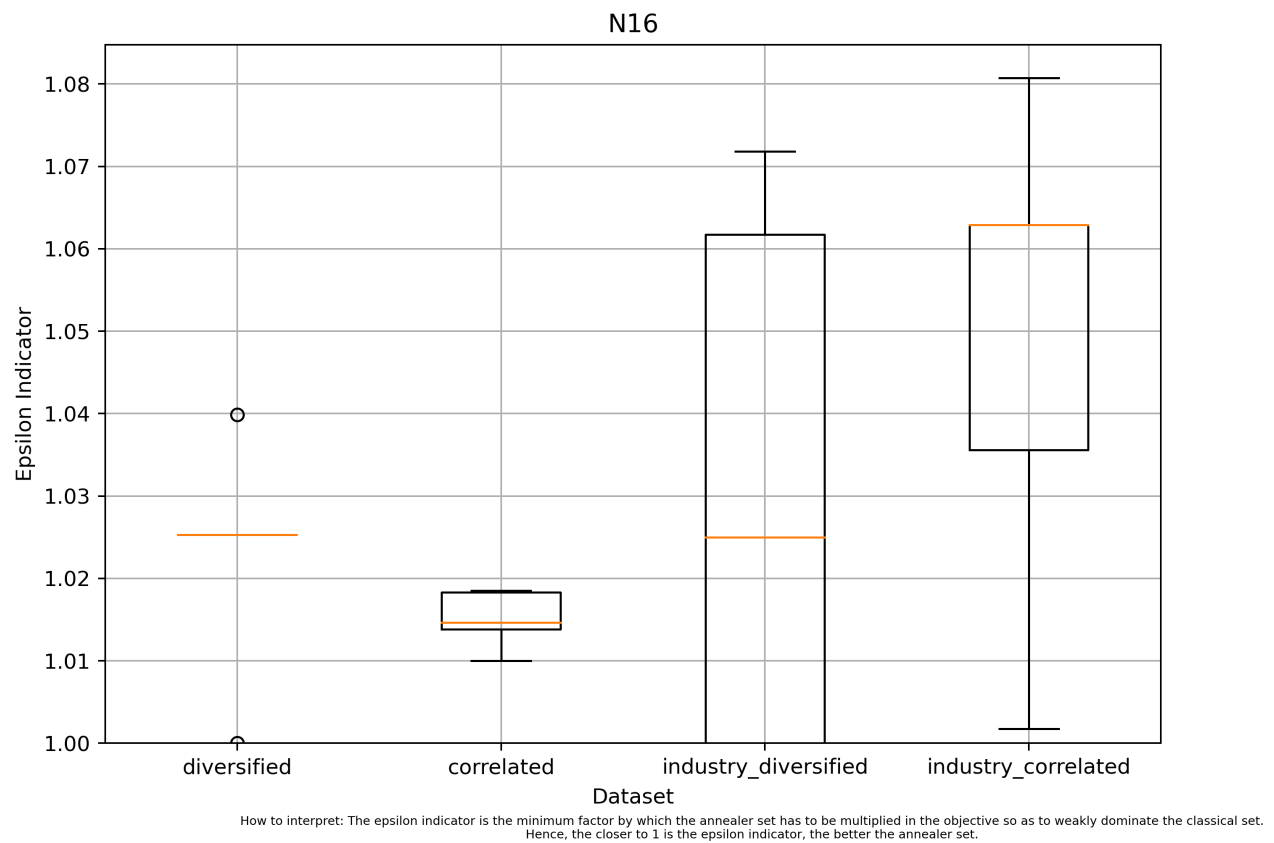
We noticed that for **N=32**, **pause** clearly outperformed the other schedules.

Scenario A3 - Datasets

For this scenario, we will study the influence from the dataset. Previous scenarios used a **industry_diversified** dataset, with assets from different industries. Therefore, we are going to introduce another dataset, called **industry_correlated**, from the same source, however, with assets from the same industry. Moreover, we will also introduce two datasets that are statistically correlated and diversified, **diversified** and **correlated**.

The results are executed for sizes **N=16**, **N=32**, and **N=64**, with parameters **chain_strength = 1.000 * maxAbs**, **B=0.5**, **mediumDmediumS** q values, **layout** embedding, and **default** schedule.

Boxplots - scenarioA3Y2021M05D09h22m11s08



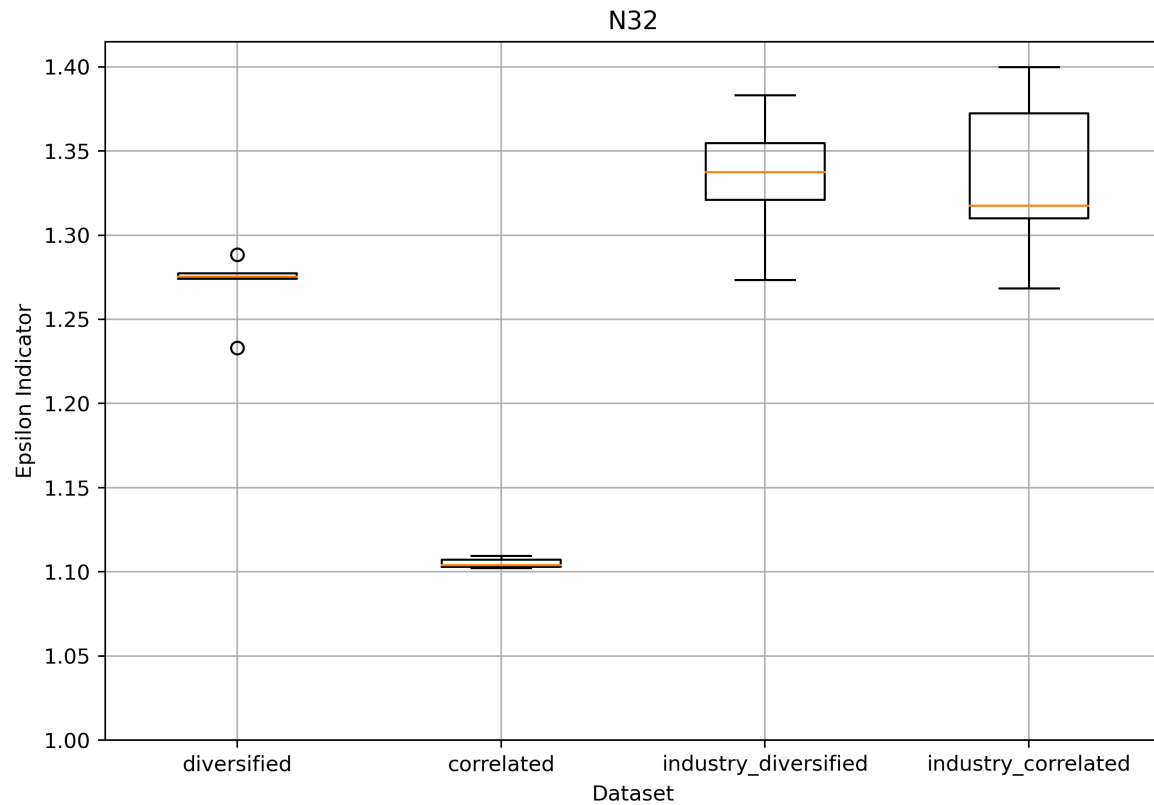
Results of Kruskal-Wallis One-way ANOVA:

H statistic: 4.251627554882673

p-value: 0.23554355060966134

The null hypothesis was not rejected!

Boxplots - scenarioA3Y2021M05D09h22m10s33



Results of Kruskal-Wallis One-way ANOVA:

H statistic: 13.217142857142846

p-value: 0.00418979564660706

Null hypothesis rejected! Now performing pairwise comparison with Dunn's test

	diversified	correlated	industry_diversified
industry_correlated			
diversified	1.000000	0.058367	0.173311
0.292077			
correlated	0.058367	1.000000	0.000402
0.000685			
industry_diversified	0.173311	0.000402	1.000000
1.000000			
industry_correlated	0.292077	0.000685	1.000000
1.000000			

diversified and correlated are NOT!

diversified and industry_diversified are NOT!

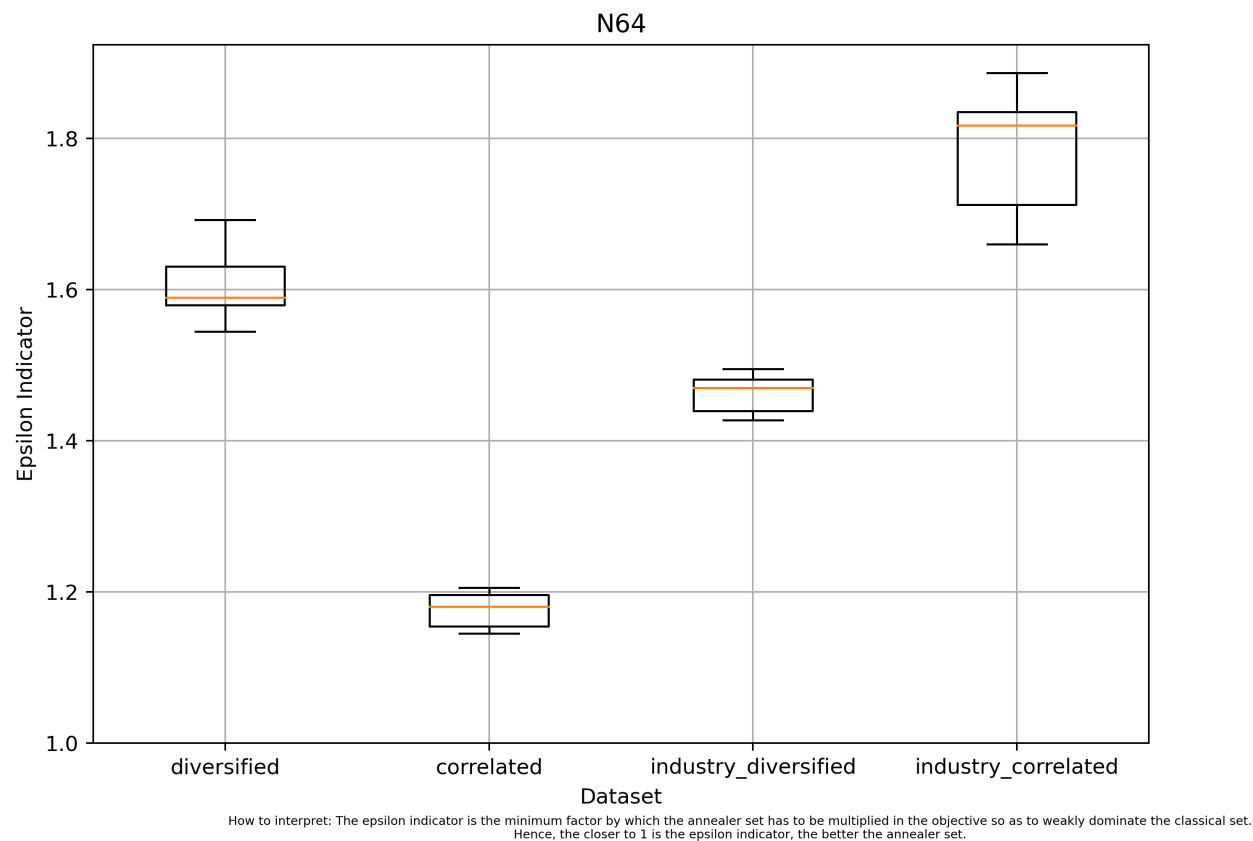
diversified and industry_correlated are NOT!

correlated and industry_diversified ARE significantly different!

correlated and industry_correlated ARE significantly different!

industry_diversified and industry_correlated are NOT!

Boxplots - scenarioA3Y2021M05D09h21m58s07



Results of Kruskal-Wallis One-way ANOVA:
H statistic: 17.582857142857137
p-value: 0.0005361519491862146
Null hypothesis rejected! Now performing pairwise comparison with Dunn's test

	diversified	correlated	industry_diversified
industry_correlated			
diversified	1.000000e+00	5.507773e-07	1.537609e-03
4.704298e-03			
correlated	5.507773e-07	1.000000e+00	2.226214e-03
2.772116e-09			
industry_diversified	1.537609e-03	2.226214e-03	1.000000e+00
9.449954e-07			
industry_correlated	4.704298e-03	2.772116e-09	9.449954e-07
1.000000e+00			

diversified and correlated ARE significantly different!
diversified and industry_diversified ARE significantly different!
diversified and industry_correlated ARE significantly different!
correlated and industry_diversified ARE significantly different!
correlated and industry_correlated ARE significantly different!
industry_diversified and industry_correlated ARE significantly different!

Key Takeaways: