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	March 30 - April 5
A3	Datasets	April 6 - April 12
B2	Embedding	April 13 - April 19
B3	Shots	April 20 - April 26
B4	Annealing	April 27 - May 3

Sidenotes to research about

- Scenario A1 epsilon values appear to follow a linear trend: `y = (x-8) * 0.0142227624 + 1`
- Find what is the maximum `N` value that is supported by dwave

Scenario A1

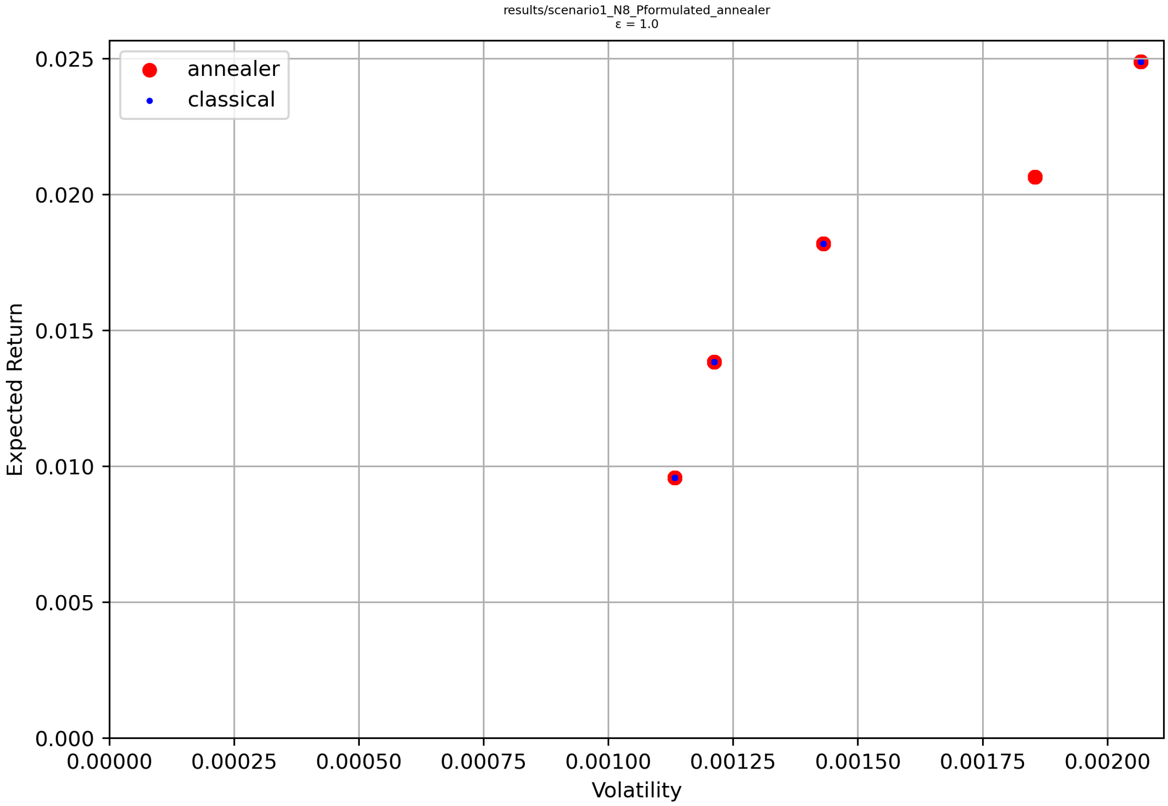
We started by experimenting several values of `N`, in order to find the maximum possible value of `N` that could be solved in a reasonable time by the classical solver.

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 and 1000 shots per execution. The `q_values` are listed in the following table:

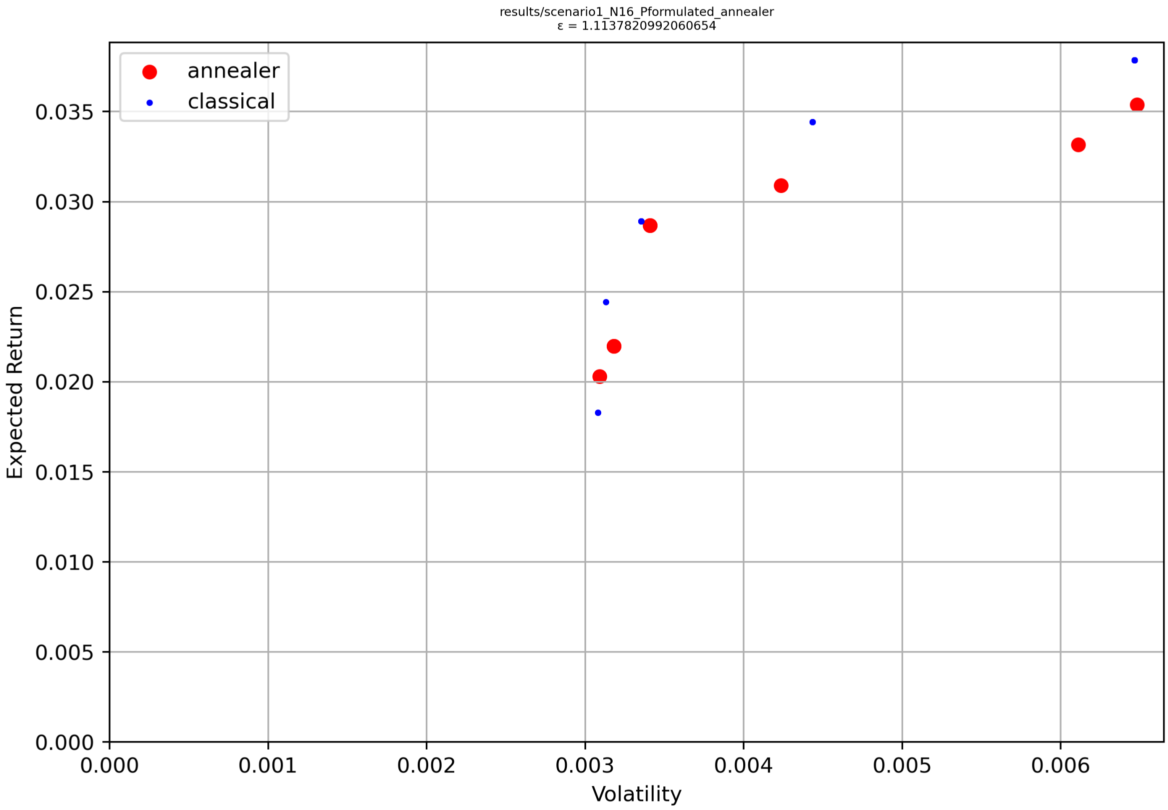
N	q values	Epsilon Indicator
8	0, 11, 20, 54	1.0
16	0, 2, 6, 100, 500	1.114
32	0, 0.4, 0.9, 2, 3, 9, 100	1.340
64	0, 0.2, 0.4, 0.6, 1.1, 1.3, 1.5, 2, 5, 6, 7, 8, 10, 100, 500	1.755

Epsilon Indicator - scenario1Y2021M03D31h18m52s49



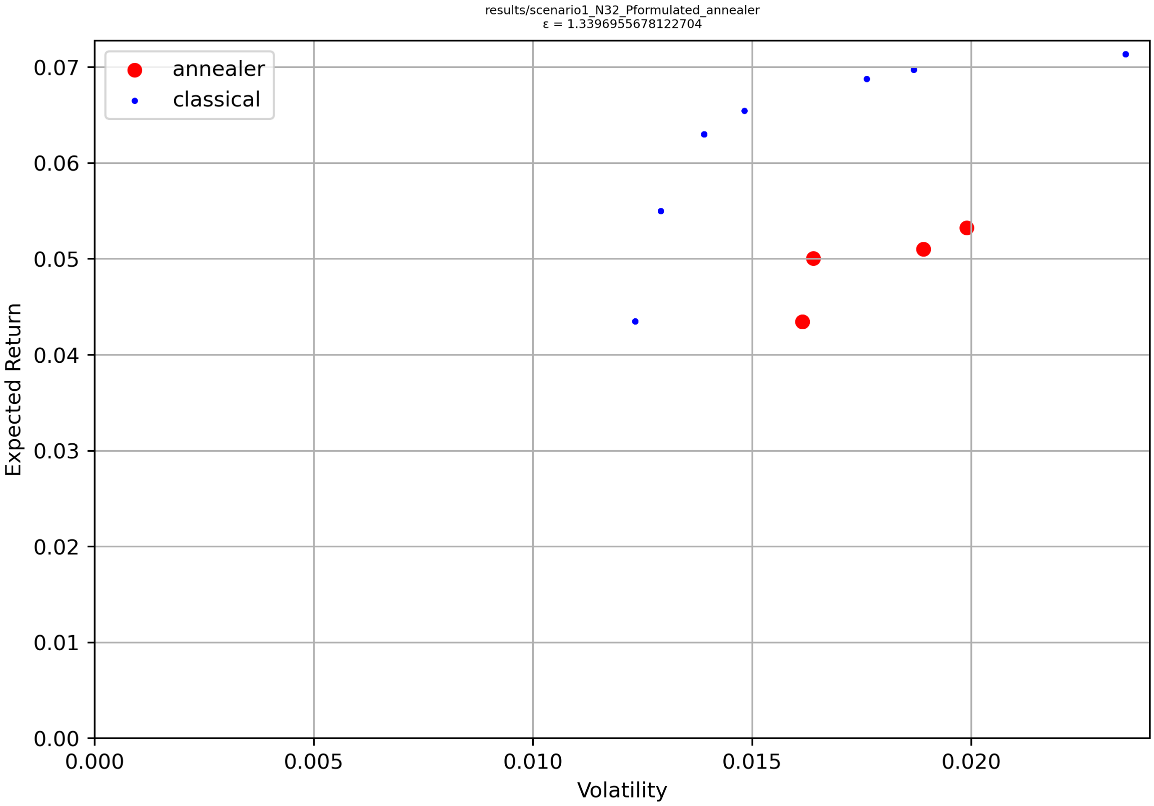
How to interpret: Blue markers are part of the efficient frontier. The epsilon indicator is the minimum factor by which the red set has to be multiplied in the objective so as to weakly dominate the blue set. Hence, the closer to 1 is the epsilon indicator, the better the red set.

Epsilon Indicator - scenario1Y2021M03D31h18m53s02



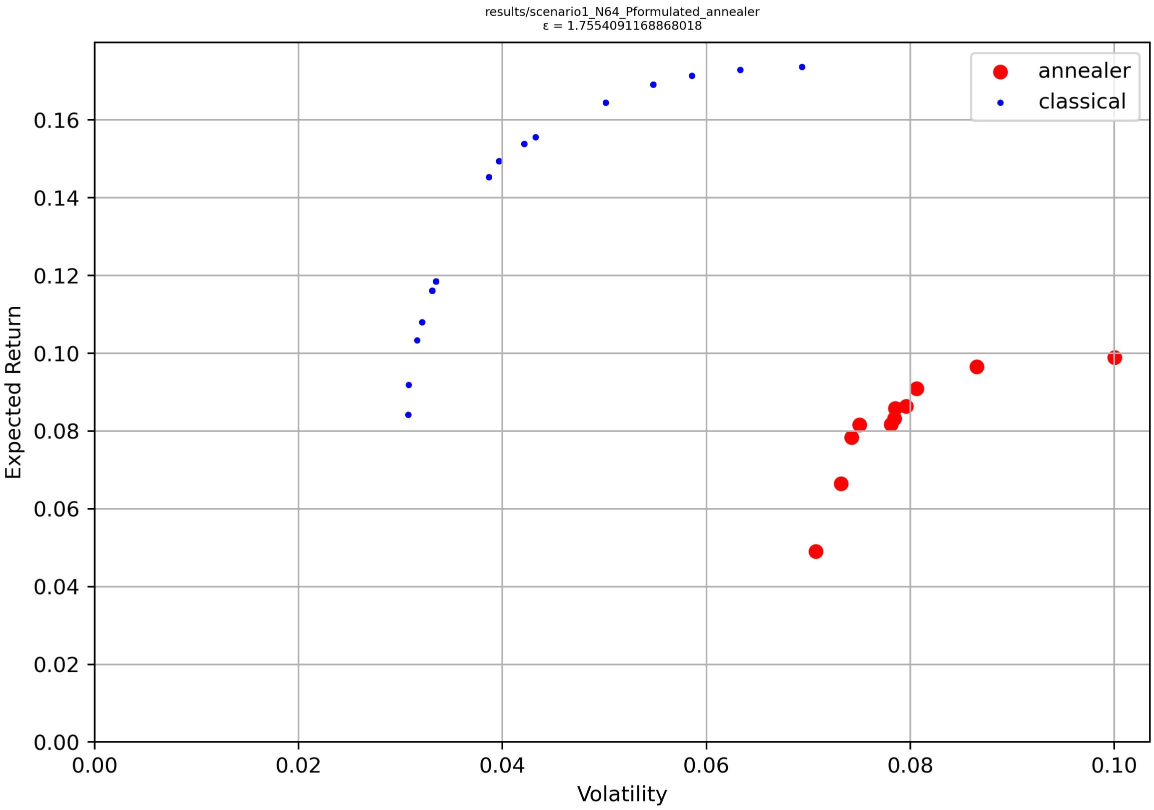
How to interpret: Blue markers are part of the efficient frontier. The epsilon indicator is the minimum factor by which the red set has to be multiplied in the objective so as to weakly dominate the blue set. Hence, the closer to 1 is the epsilon indicator, the better the red set.

Epsilon Indicator - scenario1Y2021M03D31h18m53s15



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 the blue set. Hence, the closer to 1 is the epsilon indicator, the better the red set.

Epsilon Indicator - scenario1Y2021M03D31h18m53s26



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 the blue set. Hence, the closer to 1 is the epsilon indicator, the better the red set.

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

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**
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

Chain strength	N8	N16	N32	N64
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**	**1.114**	1.340	1.755
0.125 * maxAbs	1.368	6.672	1.426	1.640
0.250 * maxAbs	**1.000**	1.167	**1.245**	1.504
0.375 * maxAbs	**1.000**	1.176*	1.275	1.429
0.500 * maxAbs	**1.000**	1.177	1.284	1.524
0.625 * maxAbs	**1.000**	1.208	1.325	**1.388**
0.750 * maxAbs	**1.000**	1.164	1.364	1.423
0.875 * maxAbs	**1.000**	1.208	1.372	1.358
1.000 * maxAbs	**1.000**	1.140	1.567	1.520
1.125 * maxAbs	**1.000**	1.182	1.350	1.421
1.250 * maxAbs	**1.000**	1.218	1.457	1.518
1.375 * maxAbs	**1.000**	1.130	1.462	1.488
1.500 * maxAbs	**1.000**	1.182	1.445	1.583

* Interestingly, this iteration got 3 of the 5 optimal solutions.

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`.

Next, for `N=64`, the sweet spot is near `chain_strength = 0.625 * maxAbs`, despite the fact that this spot does not return neither the highest fraction of valid points or the lowest fraction of chain breaks.

Finally, for `N=32`, the sweet spot is near `chain_strength = 0.250 * maxAbs`, despite the fact that this spot does not return the lowest fraction of chain breaks.

For the remaining values of `N`, no sweet spot can be accurately found. For the case `N=16`, the epsilon values are so similar that they fall under the margin of variation. For the case of `N=8`, every try gave a perfect score of `1.000`.

There is an exception for both cases of `N=8` and `N=16`. When `chain_strength = 0.125 * 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.

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

Scenario A2

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. `N=16` uses the default chain strength. `N=32` uses `chain_strength = 0.250 * maxAbs`, and `N=64` uses `chain_strength = 0.625 * maxAbs`. **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)

N	q values	Budget fraction
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)
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 (0.00131)	inf (0.31518)	1.515 (0.13468)
0.2	1.017 (0.00300)	inf (0.28815)	1.537 (0.00757)
0.3	1.026 (0.00485)	1.473 (0.11416)	1.623 (0.00443)
0.4	1.065 (0.00866)	1.222 (0.00299)	1.477 (0.00406)
0.5	1.114 (0.01153)	1.245 (0.00170)	1.388 (0.00453)
0.6	1.147 (0.00990)	1.326 (0.00115)	1.452 (0.00385)
0.7	1.275 (0.00615)	1.571 (0.00127)	2.004 (0.00389)
0.8	1.074 (0.00519)	1.195 (0.00090)	inf (0.00374)
0.9	1.032 (0.00200)	1.619 (0.00105)	inf (0.00389)

Key Takeaways:

For the case `N=16`: the chain break fraction increases until the budget is 0.5, and then decreases. Similarly, the epsilon indicator increases until 0.7 and then decreases.

For the case `N=32`: the chain break fraction is very high for the budgets 0.1, 0.2 and 0.3. After that, it gets significantly better. The epsilon indicator seems to be at its best around `B=0.5`, but the variation is too high for any meaningful conclusion.

For the case `N=64`: the chain break fraction is very high for the budget 0.1. After that, it gets significantly better. The epsilon indicator seems to be at its best around `B=0.5`. After this value, the epsilon indicator increases sharply, becoming `inf` at 0.8 and 0.9.