Quantum Error Correction on a D-Wave Quantum Annealer using 3-rep code

Muhammad Ubaid Ur Rehman

May 2025

Abstract

We investigate the impact of error-mitigation and error-correction techniques on solving the NP-hard Max-Cut problem with a D-Wave Advantage-5.4 QPU. Using a 50-node Erdős–Rényi graph, we benchmark a raw sampling baseline, built-in spin-reversal (gauge) transforms, and a custom three-repetition encoding with majority-vote decoding. Gauge transforms deliver only a modest boost over the raw baseline, whereas the Rep-3 encoding consistently yields a markedly better cut quality and lower chain-break rate. Mild parameter tuning (longer anneal windows and stronger chain strengths) further amplifies the benefit of Rep-3, highlighting a clear trade-off: lightweight built-ins give incremental gains at no qubit cost, while logical repetition offers substantial accuracy improvements at the price of increased resources.

Quantum computing holds the promise of exponentially expanding computational power, opening new frontiers in physics, chemistry, secure communications, and beyond. Yet that potential remains out of reach because today's devices are highly susceptible to noise and decoherence. Quantum error correction (QEC) encompasses the protocols and algorithms designed to suppress or mitigate these detrimental effects. Robust QEC is the key prerequisite for transforming quantum computing from a laboratory curiosity into a scalable, industry-grade technology capable of tackling real-world problems.

1 Quantum Annealing Fundamentals

Quantum annealing (QA) is a computing paradigm designed to locate low-energy configurations of an Ising or QUBO cost function by letting a quantum-mechanical system physically relax toward its ground state. Instead of executing a sequence of logic gates, the device realizes a time-dependent Hamiltonian that interpolates between a simple transverse-field driver and the user-programmed problem Hamiltonian.

1.1 Problem Hamiltonian

A quadratic unconstrained binary optimisation (QUBO) problem is mapped to an Ising Hamiltonian

$$H_P = \sum_{i=1}^{N} h_i \, \sigma_i^z + \sum_{i<j} J_{ij} \, \sigma_i^z \sigma_j^z, \tag{1}$$

where $\sigma_i^z = |0\rangle \langle 0|_i - |1\rangle \langle 1|_i$ and h_i, J_{ij} are programmable biases and couplers.

1.2 Annealing Schedule

The time-dependent Hamiltonian implemented on-chip is

$$H(t) = A(t) H_D + B(t) H_P, \qquad 0 \le t \le t_f, \tag{2}$$

with driver term $H_D = -\sum_i \sigma_i^x$. Functions $A(t)$ and $B(t)$ satisfy $A(0) \gg B(0)$ and $A(t_f) \ll B(t_f)$. For sufficiently slow total anneal time t_f, the adiabatic theorem ensures the system remains near its instantaneous ground state. Standard schedules span 1–200 μs on *Advantage* devices.

1.3 Read-out

At $t = t_f$, a dc-SQUID measurement collapses each qubit to $|0\rangle$ or $|1\rangle$. Repeating the anneal 10^3–10^4 times yields a distribution of candidate solutions.

1.4 Sources of Errors

A Thermal Noise and Excitations

During a quantum anneal the processor is held at a finite base temperature—about 20 mK on current D-Wave systems—so the qubits couple not only to the time-dependent Hamiltonian $H(t)$ but also to a thermal bath. *Thermal noise* drives stochastic single-spin and multi-spin excitations; whenever the instantaneous gap $\Delta(t)$ between the ground and first–excited state approaches $k_B T$, the bath can repopulate excited levels and thus reduce the probability of finishing the anneal in the true ground state.

B Intrinsic Control Error (ICE)

Calibration granularity and line crosstalk yield systematic deviations δh_i, δJ_{ij}, effectively perturbing H_P and biasing solutions.

$$E^{\delta}_{\text{ising}}(\mathbf{s}) \;=\; \sum_{i=1}^{N} \bigl(h_i + \delta h_i\bigr)\, s_i \;+\; \sum_{i=1}^{N} \sum_{j=i+1}^{N} \bigl(J_{i,j} + \delta J_{i,j}\bigr)\, s_i s_j,$$

C Flux Noise

$1/f$ magnetic fluctuations in Josephson junctions dephase the qubits and broaden the minimum gap.

D Chain Breaks in Minor Embeddings

Due to limitations of QPU topology, logical variables often require *chains* of ferromagnetically coupled physical qubits (minor embedding). Insufficient chain strength κ leads to broken chains; overly large κ magnifies ICE.

2 Error-Mitigation and Error-Correction Strategies

2.1 Spin-Reversal (Gauge) Transforms

For a random subset $S \subset \{1, \ldots, N\}$ we flip selected qubits and couplers,

$$h_i \;\mapsto\; (-1)^{\chi(i \in S)}\, h_i, \tag{3}$$

$$J_{ij} \;\mapsto\; (-1)^{\chi(i \in S)+\chi(j \in S)}\, J_{ij}, \tag{4}$$

where $\chi(\cdot)$ is the indicator function. Averaging over $g = 8$ gauges cancels first-order intrinsic-control error (ICE) bias with $\mathcal{O}(g)$ runtime overhead. [?]

2.2 Logical Repetition Codes

Encoding one logical spin into $k = 3$ ferromagnetically coupled qubits rescales the logical energy by k and lowers the effective temperature. Majority-vote decoding yields a logical error rate $p_{\mathrm{L}} \approx p^2$ under independent bit flips.

2.3 Chain-Strength Optimisation

The ferromagnetic penalty λ must exceed the strongest logical coupling to prevent chain breaks, but not be so large that it compresses the problem Hamiltonian. A parameter sweep over $\lambda/J_{\max} = 8 \ldots 20$ revealed a clear optimum at

$$\boxed{\lambda/J_{\max} = 16}$$

for both raw embeddings and the Rep-3 code: break-fraction drops below 0.20 while best-cut values peak as in figure below. Increasing λ beyond 16 produced no further error reduction and slightly degraded cut quality due to energy-scale compression.

2.4 Anneal-Time Schedule

We sweep four dwell times $t_{\mathrm{a}} \in \{1, 5, 10, 20\}$ µs to trade off diabatic transitions (short ramps) against thermal excitations (long ramps). The best logical energies were invariably obtained at $t_{\mathrm{a}} = 10$ µs when combined with the optimal $\lambda = 16 J_{\max}$ and eight gauges (see Fig. 3b).

2.5 Nested Quantum-Annealing Correction (NQAC)

Iterating the repetition-code construction (*nesting level* k_{N}) produces an effective Hamiltonian with a minimum gap $\propto k_{\mathrm{N}}\Delta$ at the expense of k_{N}^2 physical qubits. [?] We use $k_{\mathrm{N}} = 2$ and scale the chain strength linearly, $\lambda = k_{\mathrm{N}} \times 16 J_{\max} = 32 J_{\max}$, together with $t_{\mathrm{a}} = 20$ µs to exploit the larger gap.

3 Max-Cut Problem

Annealing is most suitable for optimization problems because the problem of finding a maximum or a minimum of a function naturally encodes in finding the ground state of the Hamiltonian corresponding to the given function.

To ground our study of quantum–error–correction (QEC) techniques we employ the *Maximum-Cut* (Max-Cut) problem as a running example. Given an undirected, weighted graph $G = (V, E, w_{ij})$, Max-Cut seeks a bipartition $V = V_A \cup V_B$ that maximises the total weight of edges crossing the cut,

$$\max_{s\in\{\pm1\}^{|V|}} \frac{1}{2} \sum_{(i,j)\in E} w_{ij}\big(1 - s_i s_j\big), \tag{5}$$

where $s_i = +1$ (-1) denotes vertex i assigned to V_A (V_B). Equation (5) is already quadratic, so it maps directly to an Ising Hamiltonian with zero local fields and pairwise couplings $J_{ij} = -w_{ij}/2$. Consequently Max-Cut can be programmed onto a D-Wave quantum-annealing QPU *without* auxiliary variables or higher-order reductions, making it an ideal vehicle for assessing hardware performance and the efficacy of QEC layers.

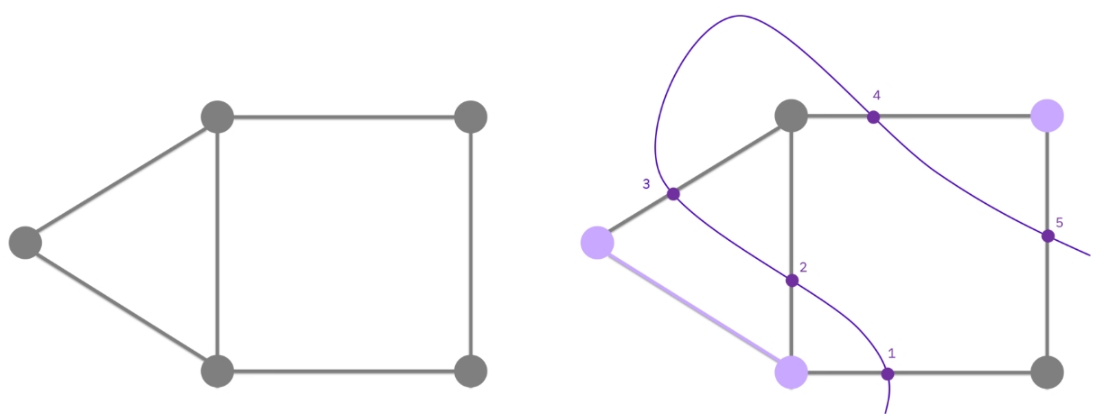


Figure 1: Max Cut Problem for a 5 node graph

Max-Cut is pertinent to error-correction research for three reasons. **(i)** For small graphs the ground-state energy is exactly known and for larger instances it can be tightly bounded, yielding an unambiguous success metric. **(ii)** Typical random or industrial graphs generate rugged energy landscapes with many near-optimal states, so thermal excitations, chain breaks, and control errors manifest clearly in the observed cut size and chain-integrity statistics. **(iii)** The logical encodings required by current QPUs are simple: ferromagnetic chains realise minor embeddings, repetition codes repair broken chains via majority vote, and spin-reversal "gauge" transforms average out first-order calibration bias. By tracking improvements in best-cut value, energy depth, and average chain-break fraction under each mitigation layer, we obtain a quantitative measure of how well present-day QEC techniques enhance solution quality on quantum annealers.

4 Experimental Results

We benchmarked five sampling pipelines on a 50-node Erdős–Rényi graph ($p = 0.8$) whose optimal Max-Cut value, established by simulated annealing (SA), is 553. Table 1 summarises the best-observed cut size, corresponding energy, and average chain-break fraction for each method.

Table 1: Headline statistics over all runs (best instance per method).

Method	**Best Cut**	**Energy**	**Break-Frac**
Sim. Ann. (CPU)	553	−115	—
Raw QPU	526	−61	0.40
Gauge QPU	523	−55	0.44
Rep-3 (no gauge)	540	−2987	0.24
Rep-3 + 2 gauges	534	−3101	0.16

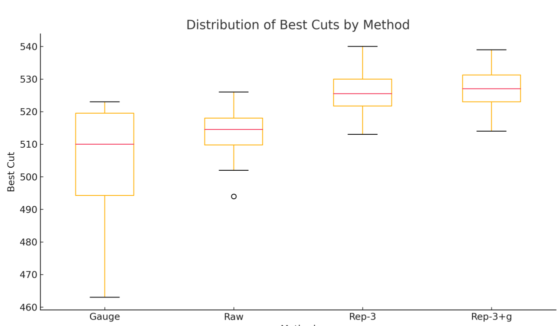
Relative to the raw embedding, spin-reversal gauges yield a modest improvement in best-cut quality, whereas the three-repetition code improves the best cut by ∼2.7% and halves the chain-break incidence.

4.1 Anneal-Time Dependence

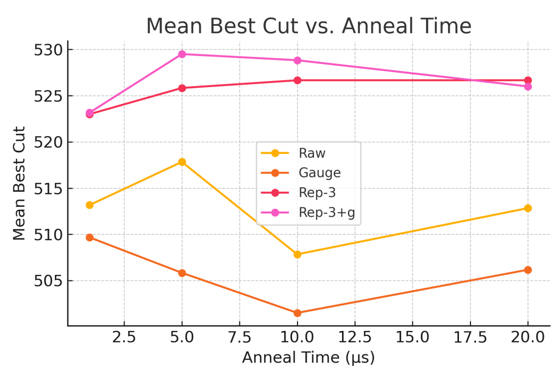
Across anneal times $t_f = 1, 5, 10, 20\,\mu$s, raw and gauge pipelines saturate near $t_f = 5\,\mu$s. In contrast, *Rep-3* continues to improve up to $t_f = 10$–$20\,\mu$s, reaching a peak cut of 540, indicating that longer schedules particularly benefit logically encoded problems.

4.2 Chain-Strength Sweep

For repetition-encoded runs we varied the ferromagnetic chain strength $\kappa \in \{1.5, 2.0, 2.5, 3.0, 12, 14, 16\}$. Performance is bimodal: $\kappa \leq 3$ leaves chains fragile (break-frac >0.60), whereas $\kappa \geq 14$ both stabilises chains and deepens energies, with an optimum at $\kappa = 16$.



(a) Distribution of best cuts.



(b) Mean best cut vs. anneal time.

Figure 2: Performance comparison of mitigation layers.

References

[1] K. L. Pudenz, T. Albash, and D. A. Lidar, "Error-corrected quantum annealing with hundreds of qubits," *Nature Communications*, vol. 5, p. 3243, 2014.

[2] K. L. Pudenz, T. Albash, and D. A. Lidar, "Quantum annealing correction for random Ising problems," *Nature Communications*, vol. 7, p. 1243, 2016.

[3] J. Marshall, V. Martin-Mayor, and I. Hen, "Analog errors in quantum annealing: doom and hope," *npj Quantum Information*, vol. 5, p. 52, 2019.

[4] M. Benedetti *et al.*, "Benchmarking quantum optimisation for the maximum-cut problem," *arXiv preprint*, 2024.

[5] E. Glauser, Z. Zhu, and H. G. Katzgraber, "Scaling advantage in approximate optimisation with quantum annealing," *Physical Review Letters*, vol. 134, p. 160601, 2025.

[6] A. D. King *et al.*, "Performance of quantum-annealing-inspired algorithms for Max-Cut," *Communications Physics*, vol. 7, p. 123, 2024.

[7] T. Stollenwerk and et al., "Benchmarking quantum annealers with near-optimal minor embeddings," *arXiv preprint*, 2024.

[8] A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics*, vol. 2, p. 5, 2014.

[9] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse ising model," *Physical Review E*, vol. 58, no. 5, pp. 5355–5363, 1998.

[10] I. Hen, J. Job, T. Albash, T. F. Rønnow, M. Troyer, and D. A. Lidar, "Probing for quantum speedup in spin glasses," *Physical Review A*, vol. 92, p. 042325, 2015.

[11] K. L. Pudenz and D. A. Lidar, "Quantum annealing correction for degenerate ground states," *Quantum Information Processing*, vol. 15, pp. 3365–3380, 2016.

[12] T. Albash, V. Martin-Mayor, I. Hen, and D. A. Lidar, "Consistency tests of classical and quantum models for a quantum annealer," *Nature Communications*, vol. 7, p. 1246, 2015.

[13] D. A. Lidar, "Towards fault-tolerant adiabatic quantum computation," in *Quantum Error Correction*, pp. 649–676, Cambridge University Press, 2018.

[14] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[15] D-Wave Systems Inc., "D-Wave Documentation Portal: Ocean SDK and Quantum Annealing." https://docs.dwavequantum.com/en/latest/, 2025. Accessed 13 May 2025.

A Raw Experimental Data

The full set of 96 measurements from six runs is provided in `raw_data.csv` and rendered below for convenience.

anneal“˙time	method	gauges	best“˙cut	energy	break“˙frac
1	Raw QPU	0	506	-21.0	0.453820
1	Gauge QPU	4	514	-65.0	0.421106
1	Rep-3	0	514	-2629.0	0.300523
1	Rep-3+2g	2	524	-2837.0	0.279958
5	Raw QPU	0	516	-41.0	0.440970
5	Gauge QPU	4	514	-37.0	0.460503
5	Rep-3	0	530	-2797.0	0.232640
5	Rep-3+2g	2	526	-2949.0	0.230478
10	Raw QPU	0	503	-15.0	0.478287
10	Gauge QPU	4	523	-55.0	0.436978
10	Rep-3	0	538	-3077.0	0.174470
10	Rep-3+2g	2	523	-2741.0	0.211593
20	Raw QPU	0	510	-29.0	0.458673
20	Gauge QPU	4	521	-51.0	0.445750
20	Rep-3	0	533	-3005.0	0.222167
20	Rep-3+2g	2	535	-3111.0	0.235960
1	Raw QPU	0	514	-37.0	0.428947
1	Gauge QPU	4	523	-55.0	0.416722
1	Rep-3	0	523	-2419.0	0.362263
1	Rep-3+2g	2	518	-2635.0	0.330020
5	Raw QPU	0	508	-25.0	0.494398
5	Gauge QPU	4	487	-47.0	0.444535
5	Rep-3	0	513	-2643.0	0.295707
5	Rep-3+2g	2	527	-2967.0	0.203515
10	Raw QPU	0	514	-37.0	0.488382
10	Gauge QPU	4	481	-57.0	0.434868
10	Rep-3	0	517	-2741.0	0.268270
10	Rep-3+2g	2	531	-2901.0	0.209395
20	Raw QPU	0	510	-29.0	0.419108
20	Gauge QPU	4	511	-59.0	0.435904
20	Rep-3	0	517	-2681.0	0.274393
20	Rep-3+2g	2	532	-2975.0	0.185080
1	Raw QPU	0	502	-13.0	0.452513
1	Gauge QPU	4	495	-35.0	0.433201
1	Rep-3	0	525	-2861.0	0.258680
1	Rep-3+2g	2	523	-2715.0	0.317792
5	Raw QPU	0	519	-47.0	0.436056
5	Gauge QPU	4	519	-47.0	0.437980
5	Rep-3	0	533	-2937.0	0.351743
5	Rep-3+2g	2	531	-3039.0	0.237393

anneal“˙time	method	gauges	best“˙cut	energy	break“˙frac
10	Raw QPU	0	494	3.0	0.466166
10	Gauge QPU	4	518	-45.0	0.376642
10	Rep-3	0	527	-2809.0	0.299003
10	Rep-3+2g	2	534	-3093.0	0.220520
20	Raw QPU	0	515	-39.0	0.465294
20	Gauge QPU	4	492	-57.0	0.463549
20	Rep-3	0	533	-2817.0	0.200023
20	Rep-3+2g	2	533	-2981.0	0.137265
1	Raw QPU	0	517	-43.0	0.402878
1	Gauge QPU	4	495	-39.0	0.465725
1	Rep-3	0	522	-2625.0	0.311367
1	Rep-3+2g	2	531	-2831.0	0.197182
5	Raw QPU	0	520	-49.0	0.397336
5	Gauge QPU	4	504	-49.0	0.453978
5	Rep-3	0	521	-2697.0	0.287337
5	Rep-3+2g	2	529	-2897.0	0.266953
10	Raw QPU	0	510	-29.0	0.487089
10	Gauge QPU	4	519	-51.0	0.423900
10	Rep-3	0	524	-2837.0	0.201440
10	Rep-3+2g	2	525	-2825.0	0.279003
20	Raw QPU	0	515	-39.0	0.457306
20	Gauge QPU	4	506	-59.0	0.458010
20	Rep-3	0	526	-2949.0	0.235223
20	Rep-3+2g	2	527	-2885.0	0.219373
1	Raw QPU	0	518	-45.0	0.435928
1	Gauge QPU	4	509	-57.0	0.454819
1	Rep-3	0	525	-2793.0	0.327070
1	Rep-3+2g	2	526	-2843.0	0.250287
5	Raw QPU	0	526	-61.0	0.401104
5	Gauge QPU	4	522	-53.0	0.426683
5	Rep-3	0	530	-2921.0	0.237527
5	Rep-3+2g	2	534	-2869.0	0.220178
10	Raw QPU	0	511	-31.0	0.417408
10	Gauge QPU	4	463	-57.0	0.443013
10	Rep-3	0	540	-2987.0	0.244513
10	Rep-3+2g	2	521	-2859.0	0.192912
20	Raw QPU	0	509	-27.0	0.415228
20	Gauge QPU	4	486	-49.0	0.440829
20	Rep-3	0	524	-2913.0	0.198520
20	Rep-3+2g	2	515	-2453.0	0.283612
1	Raw QPU	0	522	-53.0	0.420740
1	Gauge QPU	4	522	-53.0	0.452678
1	Rep-3	0	529	-2743.0	0.302290
1	Rep-3+2g	2	517	-2397.0	0.288912
5	Raw QPU	0	518	-45.0	0.463715

anneal“˙time	method	gauges	best“˙cut	energy	break“˙frac
5	Gauge QPU	4	489	-27.0	0.471148
5	Rep-3	0	528	-2685.0	0.244247
5	Rep-3+2g	2	530	-2715.0	0.210243
10	Raw QPU	0	515	-39.0	0.443601
10	Gauge QPU	4	505	-47.0	0.454340
10	Rep-3	0	514	-2643.0	0.276973
10	Rep-3+2g	2	539	-3101.0	0.155048
20	Raw QPU	0	518	-45.0	0.402633
20	Gauge QPU	4	521	-51.0	0.420207
20	Rep-3	0	527	-2883.0	0.266763
20	Rep-3+2g	2	514	-2539.0	0.243115

B Code

```
import dimod, networkx as nx
from dwave.system import DWaveSampler, EmbeddingComposite
from dwave.preprocessing import SpinReversalTransformComposite
import pandas as pd

#


# Helpers
#


def encode_repetition3(bqm, chain_strength):
    enc = dimod.BinaryQuadraticModel.empty(dimod.SPIN)
    mapping = {}
    for v in bqm.variables:
        reps = [f"{v}_r{i}" for i in range(3)]
        mapping[v] = reps
        for p in reps:
            enc.add_variable(p, bqm.linear[v])
        for i in range(3):
            for j in range(i+1, 3):
                enc.add_interaction(reps[i], reps[j], -chain_strength)
    for (u, v), J in bqm.quadratic.items():
        for pu in mapping[u]:
            for pv in mapping[v]:
                enc.add_interaction(pu, pv, J)
    return enc, mapping

def majority_vote(sample, mapping):
    return {log: 1 if sum(sample[p] for p in chain) >= 0 else -1
            for log, chain in mapping.items()}

def cut_size(sample, edges):
```

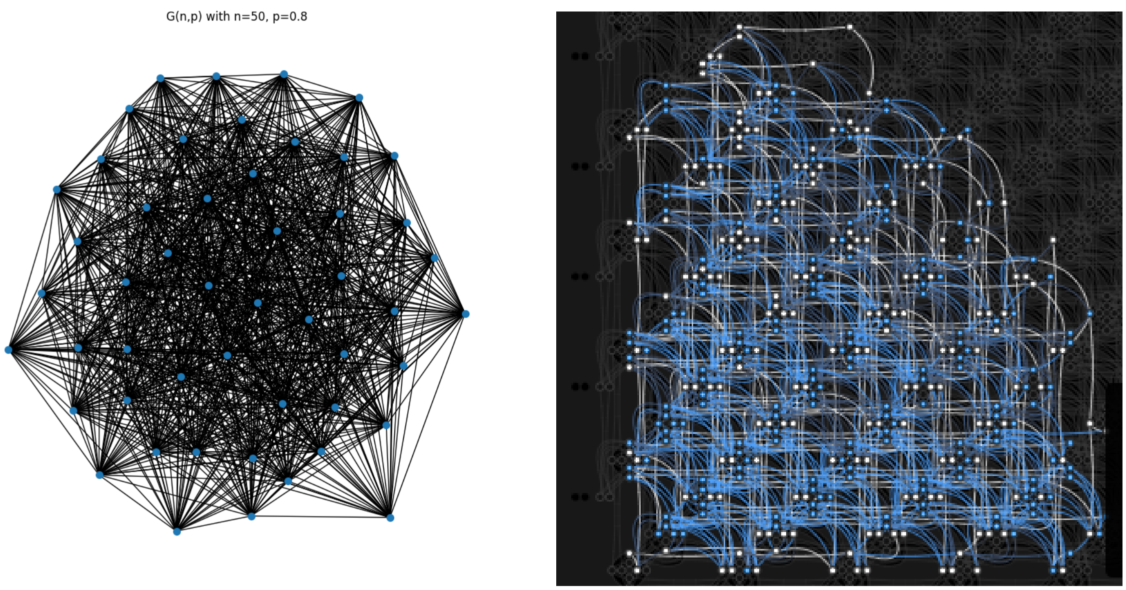
```
    return sum(sample[u] != sample[v] for u, v in edges)

def summarize_breaks(ss):
    ctx = ss.info.get("embedding_context", {})
    if "chain_break_fraction" in ctx:
        frac = ctx["chain_break_fraction"]
    elif "chain_break_fraction" in ss.record.dtype.names:
        frac = ss.record["chain_break_fraction"]
    else:
        return None, None
    return float(frac.mean()), int((frac == 0).sum())

#

# 1. Build random Max-Cut (n=50, p=0.8)
#

n, p, seed = 50, 0.8, 123
G = nx.gnp_random_graph(n, p, seed=seed)
for u, v in G.edges:
    G[u][v]["weight"] = 1

bqm = dimod.BinaryQuadraticModel.empty(dimod.SPIN)
for u, v in G.edges:
    bqm.add_interaction(u, v, +1)

#

# 2. Sampler & configs (1  s  anneal)
#

base = EmbeddingComposite(DWaveSampler())

configs = [
    {"label":"Raw␣QPU",   "reads":1000, "gauges":0, "rep":False},
    {"label":"Gauge␣QPU", "reads":1000, "gauges":4, "rep":False},
    {"label":"Rep-3",     "reads":2000, "gauges":0, "rep":True},
    {"label":"Rep-3+2g",  "reads":2000, "gauges":2, "rep":True},
]

results = []

#

# 3. Sweep annealing_time for all four methods at cs=16
#
```

```python
anneal_times = [1, 5, 10, 20]  # s

results = []
for t in anneal_times:
    print(f"\n=== annealing_time = {t} s ===")
    for cfg in configs:
        label, reads, gauges, use_rep = cfg.values()

        # 1) choose or encode problem
        if use_rep:
            problem, mapping = encode_repetition3(bqm, chain_strength=16)
        else:
            problem, mapping = bqm, None

        # 2) wrap sampler in gauges if needed
        sampler = SpinReversalTransformComposite(base) if gauges else base

        # 3) sample with current anneal time
        params = {"num_reads": reads, "annealing_time": t}
        if gauges:
            params["num_spin_reversal_transforms"] = gauges

        ss = sampler.sample(problem, **params)

        # 4) decode best sample
        phys    = ss.first.sample
        logical = majority_vote(phys, mapping) if use_rep else phys

        # 5) compute metrics
        best_cut, energy = cut_size(logical, G.edges), ss.first.energy
        brk, perfect     = summarize_breaks(ss)

        print(f"{label:12} | reads={reads:4} | best_cut={best_cut:3} "
              f"| energy={energy:6} | break-frac={brk:.3f} | perfect={perfect}")

        # also collect into table
        results.append({
            "anneal_time":  t,
            "method":       label,
            "reads":        reads,
            "gauges":       gauges,
            "repetition":   use_rep,
            "best_cut":     best_cut,
            "energy":       energy,
            "break_frac":   brk,
            "perfect_reads":perfect
        })

```

```
#
# 4. Show consolidated results table
#
import pandas as pd
df = pd.DataFrame(results)
print("\nFull results:")
print(df.to_string(index=False))
```

Listing 1: Python code that runs over multiple methods and anneal times

C Minor Embedding



(a) 50 Node Graph

(b) Corresponding minor embedding