Université Paris-Saclay,
CEA, LIST

G-score, the Maximum Cardinality Matching G_n series as a benchmark for quantum computers

Benchmark for quantum optimization heuristics

S. Louise

stephane.louise@cea.fr

Dec, 1-3, 1st QUEST-IS Conference 2025, Palaiseau

Section 1

Context

French QC applicative benchmarking “BACQ”

A polyvalent suite of benchmark for applications of QC addressing several fields

Simulation of Quantum Physics
(Quantum phase transition: Ising model in transverse field, 2D Hubbard model)

Optimization
(including Q-Score/MAXCUT)

Linear Systems Solving

Prime factorization

French QC applicative benchmarking "BACQ"

A polyvalent suite of benchmark for applications of QC addressing several fields

Simulation of Quantum Physics
(Quantum phase transition: Ising model in transverse field, 2D Hubbard model)

Optimization
(including Q-Score/MAXCUT)

Linear Systems Solving

Prime factorization

We aim to address all possible kind of QC harware, including:

- gate-based quantum computers
- analog quantum computers and quantum annealers
- work on "quantum inspired" solutions (e.g. NEC, Fujitsu ...) as reference points

The quantum utility gap & its evaluation

"*Quantum Utility*" is the current goal to achieve in Quantum Computing (QC)

- Def: *Quantum Utility* is achieved when for at least one user-relevant application, the use of QC becomes favorable over classical computing

The quantum utility gap & its evaluation

"*Quantum Utility*" is the current goal to achieve in Quantum Computing (QC)

- Def: *Quantum Utility* is achieved when for at least one user-relevant application, the use of QC becomes favorable over classical computing

Measuring the gap to *Quantum Utility* requires:

- Probing the performance of the best known Quantum Algorithms on a set of applications
- Probing the performance of the best known Classical Algorithms on the same set of applications

The quantum utility gap & its evaluation

"*Quantum Utility*" is the current goal to achieve in Quantum Computing (QC)

- Def: *Quantum Utility* is achieved when for at least one user-relevant application, the use of QC becomes favorable over classical computing

Measuring the gap to *Quantum Utility* requires:

- Probing the performance of the best known Quantum Algorithms on a set of applications
- Probing the performance of the best known Classical Algorithms on the same set of applications

For the BACQ benchmarking framework:

- Benchmarking classical computing on the same protocol provides a baseline
- The ratio between performance of QC and Classical Computing is a KPI

The quantum utility gap & its evaluation

"*Quantum Utility*" is the current goal to achieve in Quantum Computing (QC)

- Def: *Quantum Utility* is achieved when for at least one user-relevant application, the use of QC becomes favorable over classical computing

Measuring the gap to *Quantum Utility* requires:

- Probing the performance of the best known Quantum Algorithms on a set of applications
- Probing the performance of the best known Classical Algorithms on the same set of applications

For the BACQ benchmarking framework:

- Benchmarking classical computing on the same protocol provides a baseline
- The ratio between performance of QC and Classical Computing is a KPI
- N.B.: such approaches encompasses the capabilities of the hardware and the software tools

The quantum utility gap & its evaluation

"*Quantum Utility*" is the current goal to achieve in Quantum Computing (QC)

- Def: *Quantum Utility* is achieved when for at least one user-relevant application, the use of QC becomes favorable over classical computing

Measuring the gap to *Quantum Utility* requires:

- Probing the performance of the best known Quantum Algorithms on a set of applications
- Probing the performance of the best known Classical Algorithms on the same set of applications

For the BACQ benchmarking framework:

- Benchmarking classical computing on the same protocol provides a baseline
- The ratio between performance of QC and Classical Computing is a KPI
- N.B.: such approaches encompasses the capabilities of the hardware and the software tools
- But this is what the end-user cares about, so it is fine

Section 2

The Gn series, a suite of constrained optimization problems

The Gn series: a hard problem for optimization heuristics

G_n series of optimization problem:

- Was designed as a hard problem for optimization heuristics [Sasaki, Hajek, 1988]
  - Demonstrated that Simulated Annealing converges exponentially slow

The Gn series: a hard problem for optimization heuristics

G_n series of optimization problem:

- Was designed as a hard problem for optimization heuristics [Sasaki, Hajek, 1988]
  - Demonstrated that Simulated Annealing converges exponentially slow
- A Maximum Cardinality Matching problem
  - Not "hard" as in NP-hard or worst
  - The optimal configuration is known by construction

The Gn series: a hard problem for optimization heuristics

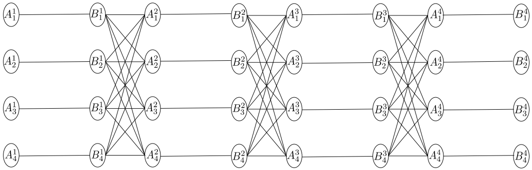
G_n series of optimization problem:

- Was designed as a hard problem for optimization heuristics [Sasaki, Hajek, 1988]
  - Demonstrated that Simulated Annealing converges exponentially slow
- A Maximum Cardinality Matching problem
  - Not "hard" as in NP-hard or worst
  - The optimal configuration is known by construction
- Interesting properties:
  - Connectivity initially low but increases with parameter n
  - The probability of choosing an optimal edge at random decreases as $\frac{1}{n+1}$

The Gn series: a hard problem for optimization heuristics

G_n series of optimization problem:

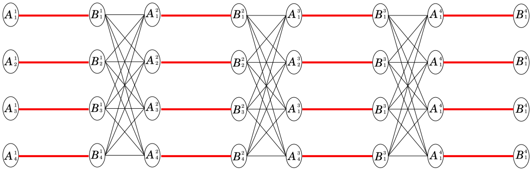
- Was designed as a hard problem for optimization heuristics [Sasaki, Hajek, 1988]
  - Demonstrated that Simulated Annealing converges exponentially slow
- A Maximum Cardinality Matching problem
  - Not "hard" as in NP-hard or worst
  - The optimal configuration is known by construction
- Interesting properties:
  - Connectivity initially low but increases with parameter n
  - The probability of choosing an optimal edge at random decreases as $\frac{1}{n+1}$



The Gn series: a hard problem for optimization heuristics

G_n series of optimization problem:

- Was designed as a hard problem for optimization heuristics [Sasaki, Hajek, 1988]
  - Demonstrated that Simulated Annealing converges exponentially slow
- A Maximum Cardinality Matching problem
  - Not "hard" as in NP-hard or worst
  - The optimal configuration is known by construction
- Interesting properties:
  - Connectivity initially low but increases with parameter n
  - The probability of choosing an optimal edge at random decreases as $\frac{1}{n+1}$



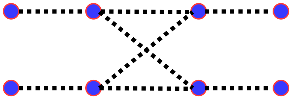
The Gn series on QA: from hard constraints to soft ones

QUBO (for minimazation):

$$q_{ee} = -1 - 2\lambda \text{ et } q_{ee'} = \begin{cases} 2\lambda & \text{si } \exists v \in N/e \in \Gamma(v) \text{ and } e' \in \Gamma(v) \\ 0 & \textit{otherwise} \end{cases}$$

As $\sum_{e \in E} x_e \leq card\{E\}$ we can decide for $\lambda = card\{E\} = (n+1)^3$ as an upper value [3]

- Example: G1



$$Q_{G_1} = \begin{bmatrix} -17 & 0 & 16 & 16 & 0 & 0 & 0 & 0 \\ 0 & -17 & 0 & 0 & 16 & 16 & 0 & 0 \\ 0 & 0 & -17 & 16 & 16 & 0 & 16 & 0 \\ 0 & 0 & 0 & -17 & 0 & 16 & 0 & 16 \\ 0 & 0 & 0 & 0 & -17 & 16 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & -17 & 0 & 16 \\ 0 & 0 & 0 & 0 & 0 & 0 & -17 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -17 \end{bmatrix}$$

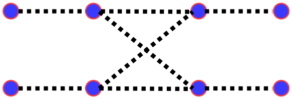
The Gn series on QA: from hard constraints to soft ones

QUBO (for minimazation):

$$q_{ee} = -1 - 2\lambda \text{ et } q_{ee'} = \begin{cases} 2\lambda & \text{si } \exists v \in N/e \in \Gamma(v) \text{ and } e' \in \Gamma(v) \\ 0 & \textit{otherwise} \end{cases}$$

As $\sum_{e \in E} x_e \leq card\{E\}$ we can decide for $\lambda = card\{E\} = (n+1)^3$ as an upper value [3]

- Example: G1



$$Q_{G_1} = \begin{bmatrix} -17 & 0 & 16 & 16 & 0 & 0 & 0 & 0 \\ 0 & -17 & 0 & 0 & 16 & 16 & 0 & 0 \\ 0 & 0 & -17 & 16 & 16 & 0 & 16 & 0 \\ 0 & 0 & 0 & -17 & 0 & 16 & 0 & 16 \\ 0 & 0 & 0 & 0 & -17 & 16 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & -17 & 0 & 16 \\ 0 & 0 & 0 & 0 & 0 & 0 & -17 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -17 \end{bmatrix}$$

- When n grows, the number of edge per vertex increases linearly
- $\Rightarrow$ the minor-embedding is rapidly an issue

Why this series of problem is interesting as a benchmark

At least a handful of papers have studied the case of G_n on various DWave QPUs and beyond:

- 2021 Initial study on DWave-2000Q [Vert et al]
- 2022 Study on DWave-Advantage + spectral gap [Mc Leod and Sasdelli]
- 2024 Full parametrization study and first Advantage-2 evaluation [Vert, Wilsch et al]
- 2025 Compared study of DWave QA with NEC's Vector Annealing (quantum inspired approach) [Louise]

Why this series of problem is interesting as a benchmark

At least a handful of papers have studied the case of G_n on various DWave QPUs and beyond:

, 2021 Initial study on DWave-2000Q [Vert et al]

, 2022 Study on DWave-Advantage + spectral gap [Mc Leod and Sasdelli]

, 2024 Full parametrization study and first Advantage-2 evaluation [Vert, Wilsch et al]

, 2025 Compared study of DWave QA with NEC's Vector Annealing (quantum inspired approach) [Louise]

This is a nice complement to Q-Score MaxCut for ATOS/EViden [1]

- MaxCut is NP-hard unconstrained optimization problem "easy" to map on Ising/QUBO

Why this series of problem is interesting as a benchmark

At least a handful of papers have studied the case of G_n on various DWave QPUs and beyond:

- 2021 Initial study on DWave-2000Q [Vert et al]
- 2022 Study on DWave-Advantage + spectral gap [Mc Leod and Sasdelli]
- 2024 Full parametrization study and first Advantage-2 evaluation [Vert, Wilsch et al]
- 2025 Compared study of DWave QA with NEC's Vector Annealing (quantum inspired approach) [Louise]

This is a nice complement to Q-Score MaxCut for ATOS/EViden [1]

- MaxCut is NP-hard unconstrained optimization problem "easy" to map on Ising/QUBO
- MCM G_n is an imperfect fit for QUBO approaches, because of hard constraints

Section 3

From a series of problem to a benchmark score

From experimental results to benchmark scores

How to evaluate the results in a platform-independent way?

From experimental results to benchmark scores

How to evaluate the results in a platform-independent way?

- Differentiating QPU results from Random results:
  - Applying a threshold to probability with statistical significance above randomness
  - For low probability results ($p \leq 10^{-4}$) the threshold is given at a significant 1% threshold (i.e. it is required to have at least 1000 non-filtrated samples)
  - For higher probability of random outcome a typical factor 10 is required, also with statistical significance.

From experimental results to benchmark scores

How to evaluate the results in a platform-independent way?

- Differentiating QPU results from Random results:
  - Applying a threshold to probability with statistical significance above randomness
  - For low probability results ($p \leq 10^{-4}$) the threshold is given at a significant 1% threshold (i.e. it is required to have at least 1000 non-filtrated samples)
  - For higher probability of random outcome a typical factor 10 is required, also with statistical significance.
- In the case of optimization, approximate solutions can be acceptable if close enough to optimal
  - Distance to optimum:
    - Hamming distance to optimum,
    - optimality score pondered by constraints violations

From experimental results to benchmark scores

How to evaluate the results in a platform-independent way?

- Differentiating QPU results from Random results:
  - Applying a threshold to probability with statistical significance above randomness
  - For low probability results ($p \leq 10^{-4}$) the threshold is given at a significant 1% threshold (i.e. it is required to have at least 1000 non-filtrated samples)
  - For higher probability of random outcome a typical factor 10 is required, also with statistical significance.
- In the case of optimization, approximate solutions can be acceptable if close enough to optimal
  - Distance to optimum:
    - Hamming distance to optimum,
    - optimality score pondered by constraints violations

In this case we decided on a mix of both Hamming distance and number of violation

- The final score = evaluation of the size of a solvable problem

Definition of G-score and weighted G- score

- The relative Hamming distance to the optimal solution

$$r_H = \frac{d_H}{(n+1)^3} \tag{1}$$

where d_H is the Hamming distance

Definition of G-score and weighted G- score

- The relative Hamming distance to the optimal solution

$$r_H = \frac{d_H}{(n+1)^3} \tag{1}$$

where d_H is the Hamming distance

- An evaluation of the optimality taking into account the invalidation of constrains into the best found solution:

$$r_d = \left(0, \frac{\mathcal{L} - f}{(n+1)^2}\right) \tag{2}$$

where $\mathcal{L}$ is the number of links in this "best" solution and f being the number of failed constraints

Definition of G-score and weighted G- score

- The relative Hamming distance to the optimal solution

$$r_H = \frac{d_H}{(n+1)^3} \tag{1}$$

where d_H is the Hamming distance

- An evaluation of the optimality taking into account the invalidation of constrains into the best found solution:

$$r_d = \left(0, \frac{\mathcal{L} - f}{(n+1)^2}\right) \tag{2}$$

where $\mathcal{L}$ is the number of links in this "best" solution and f being the number of failed constraints

- We define a ratio of optimality for the first "failed" G_n

$$r_o = \frac{1}{2}(1 - r_H + r_d) \tag{3}$$

Section 4

Definition of G-Score and Experimental results on DWave family of QPUs

G-score results

We define a weighted-score based on the probability of finding the optimal on the last "non-failed" G_n

$$S_G = (n_s + 1 + r_o \times p_o)^3 \tag{4}$$

G-score results

We define a weighted-score based on the probability of finding the optimal on the last "non-failed" G_n

$$S_G = (n_s + 1 + r_o \times p_o)^3 \tag{4}$$

Reflect the capability of solving a problem with S_G variables with constraints

G-score results

We define a weighted-score based on the probability of finding the optimal on the last "non-failed" G_n

$$S_G = (n_s + 1 + r_o \times p_o)^3 \tag{4}$$

Reflect the capability of solving a problem with S_G variables with constraints

Experimental results

QPU architecture	n_s	r_o	p_o	S_G
DWave-2000Q (Chimera)	2	0.9375	84.9%	**54.7**
DWave Advantage (Pegasus)	3	0.904	1.8%	**64.8**
DWave Advantage-2-proto (Zephir)	3	0.936	11.0%	**69.1**
SA	5	0.92	4%	220
VA2 (NEC)	7	0.93	0.7%	513
VA3 (NEC)	7	0.95	100	717

G-score results

We define a weighted-score based on the probability of finding the optimal on the last "non-failed" G_n

$$S_G = (n_s + 1 + r_o \times p_o)^3 \tag{4}$$

Reflect the capability of solving a problem with S_G variables with constraints

Experimental results

QPU architecture	n_s	r_o	p_o	S_G
DWave-2000Q (Chimera)	2	0.9375	84.9%	**54.7**
DWave Advantage (Pegasus)	3	0.904	1.8%	**64.8**
DWave Advantage-2-proto (Zephir)	3	0.936	11.0%	**69.1**
SA	5	0.92	4%	220
VA2 (NEC)	7	0.93	0.7%	513
VA3 (NEC)	7	0.95	100	717

Classical computing is still significantly better than current results of QA on the latest QPU, but QPUs are getting better

Main points and future work

- Next year, the BACQ consortium expects to provide Benchmark scores on a small subset of current QCs
  - Methodology, code and instances will be free to use to improve on the scores
  - Methods to implement and port to other QPU will be provided in preliminary format
  - We are open to discuss the relevance and improvements with tech providers or users
- We can already measure the progress from one generation to the other with this score

Main points and future work

- Next year, the BACQ consortium expects to provide Benchmark scores on a small subset of current QCs
  - Methodology, code and instances will be free to use to improve on the scores
  - Methods to implement and port to other QPU will be provided in preliminary format
  - We are open to discuss the relevance and improvements with tech providers or users
- We can already measure the progress from one generation to the other with this score

Future work:

- Porting to other analog QCs:
  - Work should start in 2026-Q1 for Pasqal QC (Ruby) and Quandela QC (Lumy)
  - End Q4-2025 to Q3-2026 for first tests on IBM QPUs
- New optimization benchmark on non quadratic problems (harder for annealers)

Acknowledgments and references

- This study was funded by the MetriQs/BACQ program under the France 2030/National Strategy for Quantum Technology initiative. MetriQs France 2030, ANR-22-QMET-0002.

References:

1. Martiel, S., Ayral, T., Allouche, C.: *Benchmarking quantum copro- cessors in an application-centric, hardware-agnostic, and scalable way.* IEEE Transactions on Quantum Engineering 2, 1–11 (2021)
2. Sasaki, G.H., Hajek, B.: *The time complexity of maximum matching by simulated annealing*. Journal of the ACM (JACM) 35(2), 387–403 (1988)
3. Vert, D., Sirdey, R., Louise, S.: *Benchmarking Quantum Annealing Against 'Hard'' Instances of the Bipartite Matching Problem*. SN Com- puter Science 2(2), 106 (Feb 2021). https://doi.org/10.1007/s42979-021- 00483-1
4. CR McLeod, M Sasdelli, *Benchmarking D-Wave quantum annealers: spectral gap scaling of maximum cardinality matching problems*. ICCS, 2022 - Springer
5. Vert, D., Willsch, M., Yenilen, B., Sirdey, R., Louise, S., Michielsen, K.: *Benchmarking quantum annealing with maximum cardinality matching problems*. Frontiers Comput. Sci. 6 (2024). https://doi.org/10.3389/FCOMP.2024.1286057, https://doi.org/10.338 9/fcomp.2024.1286057
6. Barbaresco, F., Rioux, L., Labreuche, C., Nowak, M., Olivier, N., Nicolazic, D., Hess, O., Guilmin, A.L., Wang, R., Sassolas, T., Louise, S., Snizhko, K., Misguich, G., Auffèves, A., Whitney, R., Vergnaud, E., Schopfer, F.: *BACQ – application-oriented benchmarks for quantum computing* (2024), https://arxiv.org/abs/2403.12205

Merci

Thanks!

CEA SACLAY
91191 Gif-sur-Yvette Cedex
France

Standard. + 33 1 69 08 60 00