Université Paris-Saclay,
CEA, LIST

Benchmarking DWave Quantum Computers within the BACQ initiative

Aspect of programming and optimizing for Quantum Annealers

S. Louise

stephane.louise@cea.fr

TQCI seminar Dec. 4th 2025

Section 1

Context

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)

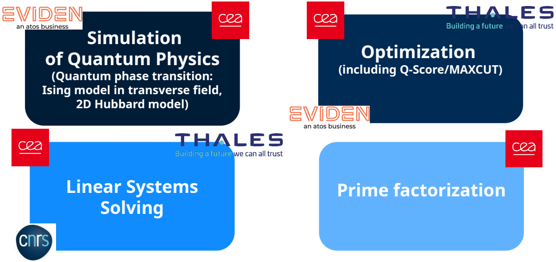
Optimization
(including Q-Score/MAXCUT)

Linear Systems Solving

Prime factorization

Context: French QC applicative benchmarking “BACQ”

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



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

- gate-based quantum computers
- analog quantum computers and quantum annealers

Application to quantum annealers (D-Wave)

D-Wave quantum annealers are a series of quantum computers:

- The oldest commercial series of quantum computers
- Not universal QCs (not gate-based)
- Primarily aimed at optimization problems (all generations) or quantum simulations (Advantage-2 only)

Application to quantum annealers (D-Wave)

D-Wave quantum annealers are a series of quantum computers:

- The oldest commercial series of quantum computers
- Not universal QCs (not gate-based)
- Primarily aimed at optimization problems (all generations) or quantum simulations (Advantage-2 only)

Utilizing machines that are not meant at universal computing make them challenging for benchmarking on general applications

Application to quantum annealers (D-Wave)

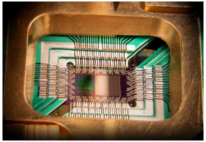
D-Wave quantum annealers are a series of quantum computers:

- The oldest commercial series of quantum computers
- Not universal QCs (not gate-based)
- Primarily aimed at optimization problems (all generations) or quantum simulations (Advantage-2 only)

Utilizing machines that are not meant at universal computing make them challenging for benchmarking on general applications

- Optimization:
  - ATOS/Eviden Q-Score (MaxCut), SotA TNO:
    - DW-2000Q **QScore= 70**
    - DW-Advantage **QScore= 180**
  - **MCM Gn series (to be presented)**
- Prime factorization:
  - SotA: DW-Advantage current best: factorization of $8,219,999 = 32,749 \times 251$
- **Linear System Solving (to be presented)**

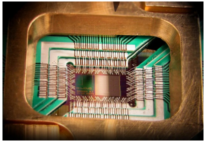
D-Wave Quantum Annealers: main points



Canadian Enterprise funded in 1999. Provider of quantum computing solutions since 2009

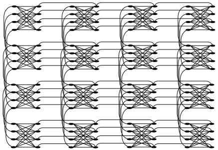
- Superconducting flux qubits (nobium)
- 5 generations of QPU: 128, 1152, 2048, 5000+
- next generation: Advantage 2, 7440 qubits (end of 2024?)
- Principle: Quantum Annealing (QA)

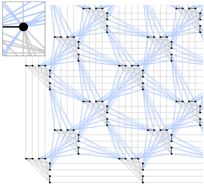
D-Wave Quantum Annealers: main points



Canadian Enterprise funded in 1999. Provider of quantum computing solutions since 2009

- Superconducting flux qubits (nobium)
- 5 generations of QPU: 128, 1152, 2048, 5000+
- next generation: Advantage 2, 7440 qubits (end of 2024?)
- Principle: Quantum Annealing (QA)



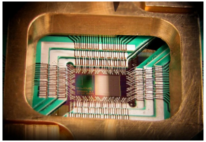


- Chimera= 6 connections/qubit
- Pegasus= 15 connections/qubit
- Zephyr= 20 connections/qubit

Really sparce compared to the number of qubits

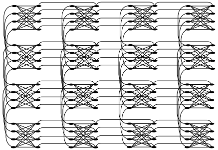
Image credits: D-Wave™

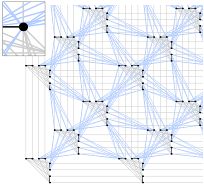
D-Wave Quantum Annealers: main points



Canadian Enterprise funded in 1999. Provider of quantum computing solutions since 2009

- Superconducting flux qubits (nobium)
- 5 generations of QPU: 128, 1152, 2048, 5000+
- next generation: Advantage 2, 7440 qubits (end of 2024?)
- Principle: Quantum Annealing (QA)





- Chimera= 6 connections/qubit
- Pegasus= 15 connections/qubit
- Zephyr= 20 connections/qubit

Really sparce compared to the number of qubits

Image credits: D-Wave™

Ising Hamiltonian

$$\mathcal{H}_{Is} = \sum_{i=0}^{n-1} h_i \sigma_i + \sum_i \sum_{j \neq i} J_{ij} \sigma_i \sigma_j \qquad \text{equiv. to QUBO problem} \qquad (1)$$

Section 2

Computing on DWave Annealers, what it means

QUBO problem, Ising Hamiltonian and Adiabatic evolution

- Generalized Ising problem (2D): $\mathcal{H}(\mathbf{h}, \mathbf{J}, \mathbf{s}) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$ with s_k spins and $J_{i,j}$ coupling constants
- QUBO problems e.g. $f = x^T Qx = \sum_{i \leq j} q_{i,j} x_i x_j$ with $x_i \in \{0, 1\}, \forall i$

QUBO problem, Ising Hamiltonian and Adiabatic evolution

- Generalized Ising problem (2D): $\mathcal{H}(\mathbf{h}, \mathbf{J}, \mathbf{s}) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$ with s_k spins and $J_{i,j}$ coupling constants
- QUBO problems e.g. $f = x^T Qx = \sum_{i \leq j} q_{i,j} x_i x_j$ with $x_i \in \{0, 1\}, \forall i$
- Variable substitutions: $s_i = 2x_i - 1$ with $s_i \in \{-1, 1\}$

QUBO problem, Ising Hamiltonian and Adiabatic evolution

- Generalized Ising problem (2D): $\mathcal{H}(\mathbf{h}, \mathbf{J}, \mathbf{s}) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$ with s_k spins and $J_{i,j}$ coupling constants
- QUBO problems e.g. $f = x^T Q x = \sum_{i \leq j} q_{i,j} x_i x_j$ with $x_i \in \{0, 1\}, \forall i$
- Variable substitutions: $s_i = 2x_i - 1$ with $s_i \in \{-1, 1\}$

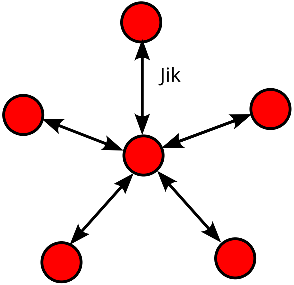
Quantum Annealing (QA) is inspired by the **Adiabatic theorem** of QM

$$\mathcal{H}(t) = f(1 - \frac{t}{\tau})\mathcal{H}_d + f(\frac{t}{\tau})\mathcal{H}_t \textbf{ with}$$
$$\begin{cases} \mathcal{H}_d = & \sum_i \sigma_i^x \\ \mathcal{H}_t = & \sum_i h_i \sigma_i^z + \sum_{(i,j) \in G} J_{i,j} \sigma_i^z \sigma_j^z \end{cases} \tag{2}$$

- $\mathcal{H}_d$ driver Hamiltonian
- $\mathcal{H}_t$ target Hamiltonian
- τ annealing time

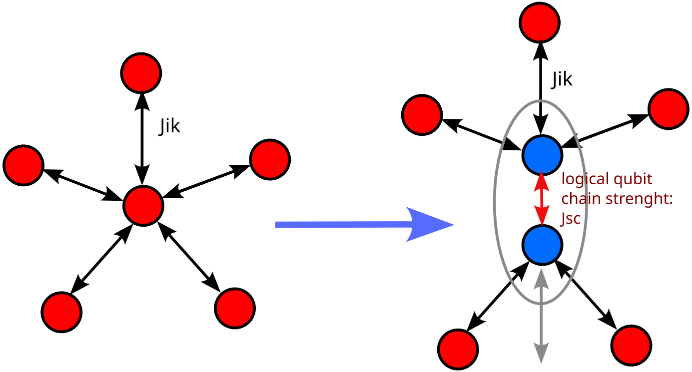
Principles and caveats of minor-embedding on D-Wave

Illustration: case of an architecture with 4 couplings/qubit and a problem with 5 couplings



Principles and caveats of minor-embedding on D-Wave

Illustration: case of an architecture with 4 couplings/qubit and a problem with 5 couplings



- Decide variable allocation and mapping (minor-embedding, NP-hard)
- Decide the chain-strenght: usually as a ratio (Relative Chain Strength, RCS)

Solving a problem on a D-Wave QPU: modus operandi

- Transform the problem into a QUBO problem or an Ising Hamiltonian
- Embed the Ising problem on the graph topology of the target QPU
  - It is utterly rare to match the topology of the DWave QPU
  - Method 1: use the hybrid solver from D-Wave
    - con: black-box, no guarantee that the QPU will be usefully taken into the work
  - Method 2: use the embedding tools of the D-Wave Ocean toolbox
    - con: quickly augments the number of qubits utilized. May fail, and rarely achieve a near optimal embedding, expect results up to $\simeq 150$ variables
- Use SRT to lower the impact of spin biases in the QPU
- Tweak the large amount of internal parameters of the QPU, notably:
  - Annealing time
  - Pause in annealing (from DW2000Q+) or reverse annealing (from Advantage+)
  - Try to obtain better embeddings and optimize the relative chain strength (RCS)
- Postprocess the results: statistical analysis, resolve logical-qubit discrepencies

Reducing biases: Spin Reversal Transform (SRT)

- The idea of SRT is to inverse a spin or negate a given variable, so:
  - For QUBO: $y_k = 1 - x_k$
  - For Ising: $\sigma'_k = -\sigma_k$

Reducing biases: Spin Reversal Transform (SRT)

- The idea of SRT is to inverse a spin or negate a given variable, so:
  - For QUBO: $y_k = 1 - x_k$
  - For Ising: $\sigma'_k = -\sigma_k$
- Why?
  - As seen current NISQ devices including D-Wave are not perfect

Reducing biases: Spin Reversal Transform (SRT)

- The idea of SRT is to inverse a spin or negate a given variable, so:
  - For QUBO: $y_k = 1 - x_k$
  - For Ising: $\sigma'_k = -\sigma_k$
- Why?
  - As seen current NISQ devices including D-Wave are not perfect
  - A non negligeable part = systemic bias
    - on qubit values
    - on couplers

Reducing biases: Spin Reversal Transform (SRT)

- The idea of SRT is to inverse a spin or negate a given variable, so:
  - For QUBO: $y_k = 1 - x_k$
  - For Ising: $\sigma'_k = -\sigma_k$
- Why?
  - As seen current NISQ devices including D-Wave are not perfect
  - A non negligeable part = systemic bias
    - on qubit values
    - on couplers
  - despite regular calibration procedures, they evolve ("slowly") through time
  - A smart utilization of SRT reduces the impact of biases

Reducing biases: Spin Reversal Transform (SRT)

- The idea of SRT is to inverse a spin or negate a given variable, so:
  - For QUBO: $y_k = 1 - x_k$
  - For Ising: $\sigma'_k = -\sigma_k$
- Why?
  - As seen current NISQ devices including D-Wave are not perfect
  - A non negligeable part = systemic bias
    - on qubit values
    - on couplers
  - despite regular calibration procedures, they evolve ("slowly") through time
  - A smart utilization of SRT reduces the impact of biases
- How?
  - **Caution**: don't change the optimal result $J'_{ik} = -J_{ik}$ et $J'_{ki} = -J_{ki}$ and $h'_k = -h_k$
    $q'_{ik} = -q_{ik}, q'_{ki} = -q_{ki}, i \neq k$ and $q'_{kk} = -q_{kk}$ et $q'_{ii} = q_{ii} + q_{ik} + q_{ki}, i \neq k$ and $C_{SRT} = q_{kk}$

Reducing biases: Spin Reversal Transform (SRT)

- The idea of SRT is to inverse a spin or negate a given variable, so:
  - For QUBO: $y_k = 1 - x_k$
  - For Ising: $\sigma'_k = -\sigma_k$
- Why?
  - As seen current NISQ devices including D-Wave are not perfect
  - A non negligeable part = systemic bias
    - on qubit values
    - on couplers
  - despite regular calibration procedures, they evolve ("slowly") through time
  - A smart utilization of SRT reduces the impact of biases
- How?
  - **Caution**: don't change the optimal result $J'_{ik} = -J_{ik}$ et $J'_{ki} = -J_{ki}$ and $h'_k = -h_k$
    $q'_{ik} = -q_{ik}, q'_{ki} = -q_{ki}, i \neq k$ and $q'_{kk} = -q_{kk}$ et $q'_{ii} = q_{ii} + q_{ik} + q_{ki}, i \neq k$ and $C_{SRT} = q_{kk}$
  - An experimentally nice rule is to split your samples in 10-20 sets with $\simeq$ 10% of random SRT in each set

Reducing biases: Spin Reversal Transform (SRT)

- The idea of SRT is to inverse a spin or negate a given variable, so:
  - For QUBO: $y_k = 1 - x_k$
  - For Ising: $\sigma'_k = -\sigma_k$
- Why?
  - As seen current NISQ devices including D-Wave are not perfect
  - A non negligeable part = systemic bias
    - on qubit values
    - on couplers
  - despite regular calibration procedures, they evolve ("slowly") through time
  - A smart utilization of SRT reduces the impact of biases
- How?
  - **Caution**: don't change the optimal result $J'_{ik} = -J_{ik}$ et $J'_{ki} = -J_{ki}$ and $h'_k = -h_k$
    $q'_{ik} = -q_{ik}, q'_{ki} = -q_{ki}, i \neq k$ and $q'_{kk} = -q_{kk}$ et $q'_{ii} = q_{ii} + q_{ik} + q_{ki}, i \neq k$ and $C_{SRT} = q_{kk}$
  - An experimentally nice rule is to split your samples in 10-20 sets with $\simeq 10\%$ of random SRT in each set

Limits on D-Wave precision, other technics

Internally, the authorized values for magnetic bias (h vector) is $\forall i, h_i \in [-2, 2]$ **and for coupling constants,** $\forall i > j, J_{i,j} \in [-1, 1]$

Limits on D-Wave precision, other technics

Internally, the authorized values for magnetic bias (h vector) is $\forall i, h_i \in [-2, 2]$ and for coupling constants, $\forall i > j, J_{i,j} \in [-1, 1]$

- **The precision on coupling or bias values is**
  - **non linear**
  - **comprised between 2% and 1‰**

Limits on D-Wave precision, other technics

Internally, the authorized values for magnetic bias (h vector) is $\forall i, h_i \in [-2, 2]$ and for coupling constants, $\forall i > j, J_{i,j} \in [-1, 1]$

- **The precision on coupling or bias values is**
  - **non linear**
  - **comprised between 2% and 1‰**
  - **my personal rule of thumb (atm): check that $\frac{\min\left(\min_{i,j}|J_{i,j}|, \min_i \left|h_i\right|\right)}{\max\left(\max_{i,j}\left|J_{i,j}\right|, \max_i \left|h_i\right|\right)} > 0.01$ for all values $J_{i,j} \neq 0$ and $h_i \neq 0$**

Limits on D-Wave precision, other technics

Internally, the authorized values for magnetic bias (h vector) is $\forall i, h_i \in [-2, 2]$ and for coupling constants, $\forall i > j, J_{i,j} \in [-1, 1]$

- **The precision on coupling or bias values is**
  - **non linear**
  - **comprised between 2% and 1‰**
  - **my personal rule of thumb (atm): check that** $\frac{\min\left(\min_{i,j}|J_{i,j}|,\min_i\left|h_i\right|\right)}{\max\left(\max_{i,j}\left|J_{i,j}\right|,\max_i\left|h_i\right|\right)} > 0.01$ **for all values** $J_{i,j} \neq 0$ **and** $h_i \neq 0$

For auxiliary variables/qubits a simple way (e.g. to circumvent the topological limitations)

- **For Ising:** $\lambda q_i q_j$
- **For QUBO:** $\lambda'(x_i + x_j - 2x_i x_j)$

Limits on D-Wave precision, other technics

Internally, the authorized values for magnetic bias (h vector) is $\forall i, h_i \in [-2, 2]$ and for coupling constants, $\forall i > j, J_{i,j} \in [-1, 1]$

- The precision on coupling or bias values is
  - non linear
  - comprised between 2% and 1‰
  - my personal rule of thumb (atm): check that $\frac{\min\left(\min_{i,j}|J_{i,j}|, \min_i \left|h_i\right|\right)}{\max\left(\max_{i,j}\left|J_{i,j}\right|, \max_i \left|h_i\right|\right)} > 0.01$ for all values $J_{i,j} \neq 0$ and $h_i \neq 0$

For auxiliary variables/qubits a simple way (e.g. to circumvent the topological limitations)

- For Ising: $\lambda q_i q_j$
- For QUBO: $\lambda'(x_i + x_j - 2x_i x_j)$
- for λ we usually use $\max_{i,j}\left(|J_{i,j}|, |h_i|\right) \times r^*$ where r^* is the Relative Chain Strenght (RCS)
- for r^* a good choice is generally between 0.5 and 3
- in case of logical qubits: possible to use open or closed chains, and use majority vote when the values of L-qubits don't agree at the Ph-qubit level

Limits on D-Wave precision, other technics

Internally, the authorized values for magnetic bias (h vector) is $\forall i, h_i \in [-2, 2]$ and for coupling constants, $\forall i > j, J_{i,j} \in [-1, 1]$

- **The precision on coupling or bias values is**
  - **non linear**
  - **comprised between 2% and 1‰**
  - **my personal rule of thumb (atm): check that $\frac{\min\left(\min_{i,j}|J_{i,j}|, \min_i\left|h_i\right|\right)}{\max\left(\max_{i,j}\left|J_{i,j}\right|, \max_i\left|h_i\right|\right)} > 0.01$ for all values $J_{i,j} \neq 0$ and $h_i \neq 0$**

For auxiliary variables/qubits a simple way (e.g. to circumvent the topological limitations)

- **For Ising: $\lambda q_i q_j$**
- **For QUBO: $\lambda'(x_i + x_j - 2x_i x_j)$**
- **for λ we usually use $\max_{i,j}\left(|J_{i,j}|, |h_i|\right) \times r^*$ where r^* is the Relative Chain Strenght (RCS)**
- **for r^* a good choice is generally between 0.5 and 3**
- **in case of logical qubits: possible to use open or closed chains, and use majority vote when the values of L-qubits don't agree at the Ph-qubit level**
- **auxiliary qubits are also necessary for quadratization methods (going from HOBO to QUBO where HOBO cost function is a generic polynomial of x_i)**

Quadratization

Quadratization is a tool to transform higher order polynomial cost functions into QUBO.
Several method exist e.g.:

- Rosenberg's procedure: for any monomial of higher order,
  - choose (i, j) and create an auxiliary variable y_{ij}

Quadratization

Quadratization is a tool to transform higher order polynomial cost functions into QUBO.
Several method exist e.g.:

- Rosenberg's procedure: for any monomial of higher order,
  - choose (i, j) and create an auxiliary variable y_{ij}
  - replace $x_i x_j$ by y_{ij} in the cost function

Quadratization

Quadratization is a tool to transform higher order polynomial cost functions into QUBO.
Several method exist e.g.:

- Rosenberg's procedure: for any monomial of higher order,
  - choose (i, j) and create an auxiliary variable y_{ij}
  - replace $x_i x_j$ by y_{ij} in the cost function
  - add a new term: $M(x_i x_j - 2x_i y_{ij} - 2x_j y_{ij} + 3y_{ij})$ with $M > 0$ and large enough

Quadratization

Quadratization is a tool to transform higher order polynomial cost functions into QUBO.
Several method exist e.g.:

- Rosenberg's procedure: for any monomial of higher order,
  - choose (i, j) and create an auxiliary variable y_{ij}
  - replace $x_i x_j$ by y_{ij} in the cost function
  - add a new term: $M(x_i x_j - 2x_i y_{ij} - 2x_j y_{ij} + 3y_{ij})$ with $M > 0$ and large enough
  - repeat until a QUBO formulation is obtained

Quadratization

Quadratization is a tool to transform higher order polynomial cost functions into QUBO.
Several method exist e.g.:

- Rosenberg's procedure: for any monomial of higher order,
  - choose (i, j) and create an auxiliary variable y_{ij}
  - replace $x_i x_j$ by y_{ij} in the cost function
  - add a new term: $M(x_i x_j - 2x_i y_{ij} - 2x_j y_{ij} + 3y_{ij})$ with $M > 0$ and large enough
  - repeat until a QUBO formulation is obtained
- NTR-KZFD (Negative Term Reduction by Kolmogorov, Zabih, Freedman and Drineas)
  - con: produces more terms, necessitate some rewriting
  - pro: the coefficients are (much) smaller

Section 3

Benchmarking Optimization problems on DWave

The choice of the Gn series as a benchmarking tool

Quantum Annealing being a heuristic we should treat it as such

- Finding a combinatorial problem with a simple to evaluate optimum
- ATOS defined the Q-Score [1] based on the Max-Cut problem (NP-hard, unconstrained)

The choice of the Gn series as a benchmarking tool

Quantum Annealing being a heuristic we should treat it as such

- Finding a combinatorial problem with a simple to evaluate optimum
- ATOS defined the Q-Score [1] based on the Max-Cut problem (NP-hard, unconstrained)
- We decided on a more simple but constrained and difficult for optimization heuristics
    - the Maximum Cardinality Matching (MCM)

The choice of the Gn series as a benchmarking tool

Quantum Annealing being a heuristic we should treat it as such

- Finding a combinatorial problem with a simple to evaluate optimum
- ATOS defined the Q-Score [1] based on the Max-Cut problem (NP-hard, unconstrained)
- We decided on a more simple but constrained and difficult for optimization heuristics
    - the Maximum Cardinality Matching (MCM)

Particularly the G_n series defined by Sasaki & Hajek [2], 1988

The choice of the Gn series as a benchmarking tool

Quantum Annealing being a heuristic we should treat it as such

- Finding a combinatorial problem with a simple to evaluate optimum
- ATOS defined the Q-Score [1] based on the Max-Cut problem (NP-hard, unconstrained)
- We decided on a more simple but constrained and difficult for optimization heuristics
  - the Maximum Cardinality Matching (MCM)

Particularly the G_n series defined by Sasaki & Hajek [2], 1988

- A matching is a single bound between 2 populations (= constrained problem)
- A maximum matching is the configuration of matchings that maximize their number
- The G_n series = instances of MCM, easy to solve
  - demonstrated the slow convergence of the **Simulated Annealing** in some cases
  - $(n+1)^3$ possible matchings but only $(n+1)^2$ in the solution
  - Selecting randomly a given matching is increasingly counterproductive as n grows

The choice of the Gn series as a benchmarking tool

Quantum Annealing being a heuristic we should treat it as such

- Finding a combinatorial problem with a simple to evaluate optimum
- ATOS defined the Q-Score [1] based on the Max-Cut problem (NP-hard, unconstrained)
- We decided on a more simple but constrained and difficult for optimization heuristics
  - the Maximum Cardinality Matching (MCM)

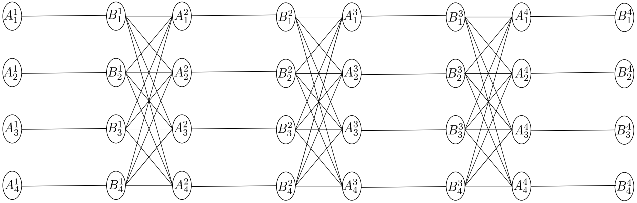
Particularly the G_n series defined by Sasaki & Hajek [2], 1988

- A matching is a single bound between 2 populations (= constrained problem)
- A maximum matching is the configuration of matchings that maximize their number
- The G_n series = instances of MCM, easy to solve
  - demonstrated the slow convergence of the **Simulated Annealing** in some cases
  - $(n+1)^3$ possible matchings but only $(n+1)^2$ in the solution
  - Selecting randomly a given matching is increasingly counterproductive as n grows

The G_n series = good candidate to complement the Q-Score on optimization

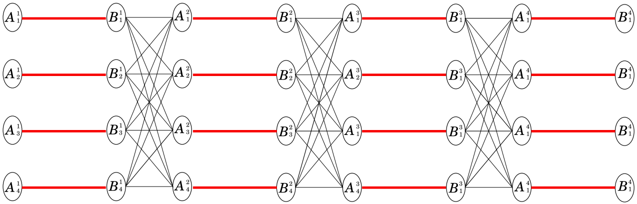
G_n series illustration

Exemple: G_3



G_n series illustration

Exemple: G_3

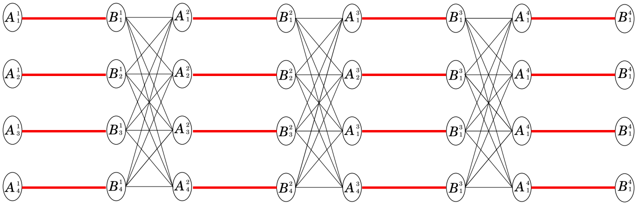


This is a “trap” for simulated annealing:

- There is a high chance to select an edge in the bipartite portions of the graph

G_n series illustration

Exemple: G_3



This is a "trap" for simulated annealing:

- There is a high chance to select an edge in the bipartite portions of the graph

We must adapt it to an Ising Hamiltonian/QUBO formulation

- The maximum matching problem is constrained $\neq$ QUBO/ISing
- Change the economic function (the Hamiltonian) to take the constraints into account
- Doing so transform the problem from hard constrained to soft constrained (good enough)

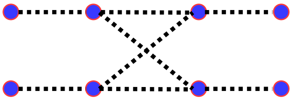
QUBO formulation

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 $\lambda = card\{E\}$ as an upper value [3]

- Example: G1



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

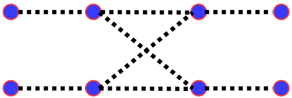
QUBO formulation

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 $\lambda = card\{E\}$ 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

Relevant parameters for improving the outcome

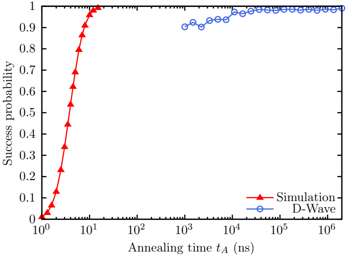
- In [3] the oldest architecture (Chimera), embedding constraints limited the achievable problems to G_4 in the best case (with at most 2000 qubits)
  - Little influence of any parameter on the outcome

Relevant parameters for improving the outcome

- In [3] the oldest architecture (Chimera), embedding constraints limited the achievable problems to G_4 in the best case (with at most 2000 qubits)
  - Little influence of any parameter on the outcome
- In [4] with JFZ, the architecture was extended to Pegasus (Advantage QPU) and Zephyr (Advantage-2 prototype)
  - We studied the influence of the annealing time τ
  - The count of qubits in the minor-embedding
  - The Relative Chain Strength (rcs) value

Relevant parameters for improving the outcome

- In [3] the oldest architecture (Chimera), embedding constraints limited the achievable problems to G_4 in the best case (with at most 2000 qubits)
  - Little influence of any parameter on the outcome
- In [4] with JFZ, the architecture was extended to Pegasus (Advantage QPU) and Zephyr (Advantage-2 prototype)
  - We studied the influence of the annealing time τ
  - The count of qubits in the minor-embedding
  - The Relative Chain Strength (rcs) value



Comparing resolution of Schrodinger eq. on supercomputer vs D-Wave experiments (Advantage-2)

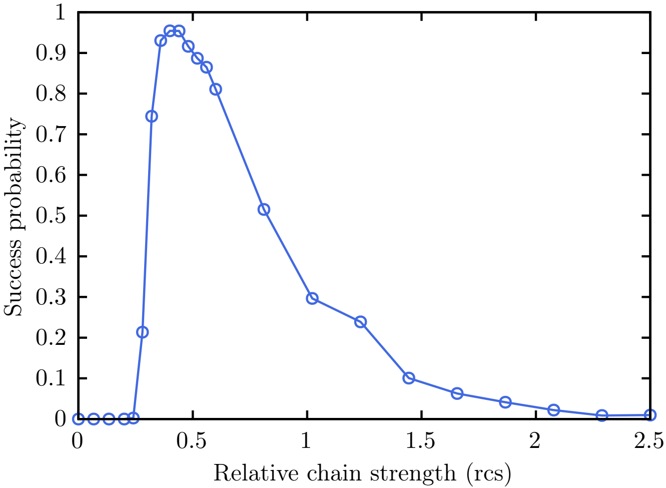
- Different from theoretical but not relevant

Section 4

Experiments on DWave familly of QPUs

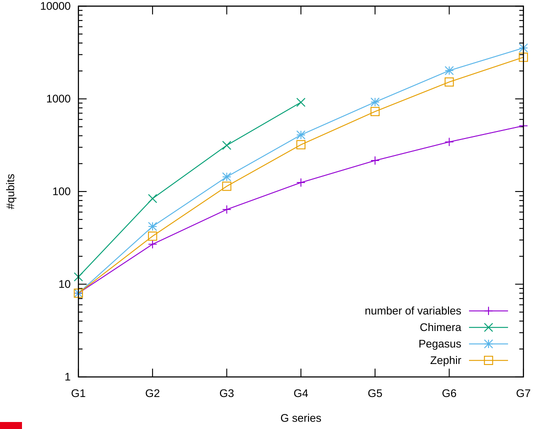
Impact of chain strength and minor-embedding

Relative Chain Strength must be well chosen to optimize the outcomes



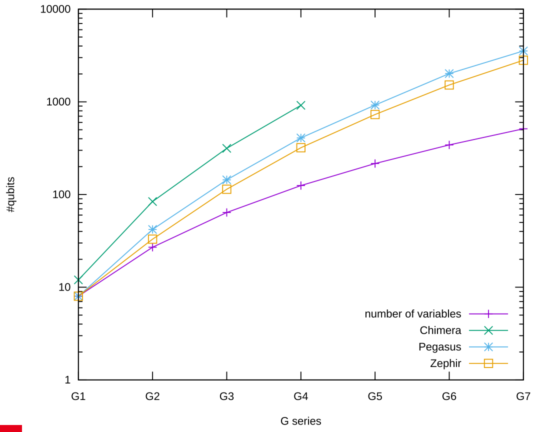
Impact of minor-embedding

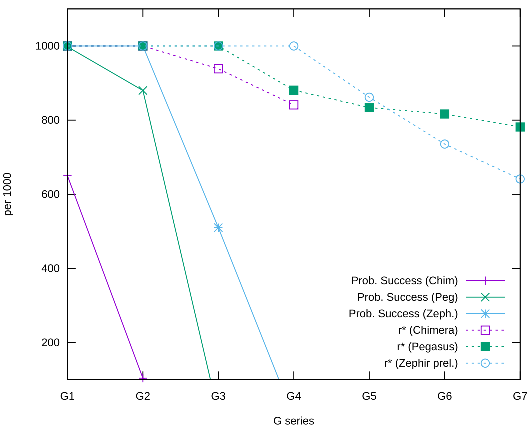
The number of qubits in the minor embedding should be reduced as much as possible to improve results



Impact of minor-embedding

The number of qubits in the minor embedding should be reduced as much as possible to improve results





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.
- Cases where an exact solution is mandatory (e.g. LSS)
  - We apply the previous criterion on the probability of correct outcome

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.
- Cases where an exact solution is mandatory (e.g. LSS)
  - We apply the previous criterion on the probability of correct outcome
- Cases where approximate solutions can be acceptable (e.g. Optimization problems)
  - Distance to optimum:
    - Hamming distance to optimum,
    - optimality score pondered by constraints violations

In the case of MCM (optimization problem we decided to mix both)

Definition of G-score

- The relative Hamming distance to the optimal solution

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

where d_H is the Hamming distance

Definition of G-score

- The relative Hamming distance to the optimal solution

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

where d_H is the Hamming distance

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

$$r_d = \left(0, \frac{\mathcal{L} - f}{(n+1)^2}\right) \tag{4}$$

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

Definition of G-score

- The relative Hamming distance to the optimal solution

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

where d_H is the Hamming distance

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

$$r_d = \left(0, \frac{\mathcal{L} - f}{(n+1)^2}\right) \tag{4}$$

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

$$r_o = \frac{1}{2}(1 - r_H + r_d) \tag{5}$$

G-score results

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

$$S_G = (n_s + 1 + r_o \times p_o)^3 \tag{6}$$

G-score results

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

$$S_G = (n_s + 1 + r_o \times p_o)^3 \tag{6}$$

It evaluates the solving capabilities of a given [quantum] optimization heuristic

G-score results

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

$$S_G = (n_s + 1 + r_o \times p_o)^3 \tag{6}$$

It evaluates the solving capabilities of a given [quantum] optimization heuristic

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 (Zephir)	3	0.936	11.0%	69.1
SA	5	0.92	4%	220
VA2	7	0.93	0.7%	513
VA3	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

$$S_G = (n_s + 1 + r_o \times p_o)^3 \tag{6}$$

It evaluates the solving capabilities of a given [quantum] optimization heuristic

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 (Zephir)	3	0.936	11.0%	69.1
SA	5	0.92	4%	220
VA2	7	0.93	0.7%	513
VA3	7	0.95	100	717

There is still a significant gap between QA and classical optimization heuristic. We can see improvements, nonetheless

Section 5

Linear System Solving

Benchmarking Linear System Solving

Linear System Solving is a highly expected application of QC:

Benchmarking Linear System Solving

Linear System Solving is a highly expected application of QC:

- Exponential speed-up (HHL algorithm, Harrow, Hassidim and Lloyd [2009])

Benchmarking Linear System Solving

Linear System Solving is a highly expected application of QC:

- Exponential speed-up (HHL algorithm, Harrow, Hassidim and Lloyd [2009])
- Potential application: numerical simulations, partial differential equation solving, ML, etc.

Benchmarking Linear System Solving

Linear System Solving is a highly expected application of QC:

- Exponential speed-up (HHL algorithm, Harrow, Hassidim and Lloyd [2009])
- Potential application: numerical simulations, partial differential equation solving, ML, etc.

HHL is not applicable to Quantum Simulators or Quantum Annealers.

Benchmarking Linear System Solving

Linear System Solving is a highly expected application of QC:

- Exponential speed-up (HHL algorithm, Harrow, Hassidim and Lloyd [2009])
- Potential application: numerical simulations, partial differential equation solving, ML, etc.

HHL is not applicable to Quantum Simulators or Quantum Annealers.

- The benchmark suite is not about Quantum Advantage
- It is about technological readiness and comparison/metrology

Benchmarking Linear System Solving

Linear System Solving is a highly expected application of QC:

- Exponential speed-up (HHL algorithm, Harrow, Hassidim and Lloyd [2009])
- Potential application: numerical simulations, partial differential equation solving, ML, etc.

HHL is not applicable to Quantum Simulators or Quantum Annealers.

- The benchmark suite is not about Quantum Advantage
- It is about technological readiness and comparison/metrology

Quantum Annealing / D-Wave quantum computers

- Are not easily comparable to gate-based QC
- Are specialized computer for Ising Hamiltonian simulations or QUBO solving
- Are not expected to reach exponential speedup (only polynomial)

State of the art: solving linear systems on D-Wave's QCs

Solving problem on D-Wave quantum annealers

To convey calculation on a D-Wave computer means finding a QUBO or Ising formulation whose minimum solves the initial problem

State of the art: solving linear systems on D-Wave's QCs

Solving problem on D-Wave quantum annealers

To convey calculation on a D-Wave computer means finding a QUBO or Ising formulation whose minimum solves the initial problem

Solving $A\mathbf{x} = \boldsymbol{b}$ where $A \in \mathcal{M}_n(\mathbf{R})$ and $\boldsymbol{b} \in \mathbb{R}^n$

$$i \in \{0, \cdots, n-1\}, \qquad \boldsymbol{A}_i.\mathbf{x} - b_i = 0$$

State of the art: solving linear systems on D-Wave's QCs

Solving problem on D-Wave quantum annealers

To convey calculation on a D-Wave computer means finding a QUBO or Ising formulation whose minimum solves the initial problem

Solving $A\mathbf{x} = \boldsymbol{b}$ where $A \in \mathcal{M}_n(\mathbf{R})$ and $\boldsymbol{b} \in \mathbb{R}^n$

$$i \in \{0, \cdots, n-1\}, \qquad \boldsymbol{A}_i.\mathbf{x} - b_i = 0$$

If we define the cost function as the least square distance to 0:

$$\mathcal{C}(\mathbf{x}) = \sum_{i=0}^{n-1} |\boldsymbol{A}_i.\mathbf{x} - b_i|^2 \tag{7}$$

State of the art: solving linear systems on D-Wave's QCs

Solving problem on D-Wave quantum annealers

To convey calculation on a D-Wave computer means finding a QUBO or Ising formulation whose minimum solves the initial problem

Solving $A\mathbf{x} = \boldsymbol{b}$ where $A \in \mathcal{M}_n(\mathbf{R})$ and $\boldsymbol{b} \in \mathbb{R}^n$

$$i \in \{0, \cdots, n-1\}, \qquad \boldsymbol{A}_i.\mathbf{x} - b_i = 0$$

If we define the cost function as the least square distance to 0:

$$\mathcal{C}(\mathbf{x}) = \sum_{i=0}^{n-1} |\boldsymbol{A}_i.\mathbf{x} - b_i|^2 \tag{7}$$

$\mathcal{C}(\mathbf{x}) \geq 0$ per definition

State of the art: solving linear systems on D-Wave's QCs

Solving problem on D-Wave quantum annealers

To convey calculation on a D-Wave computer means finding a QUBO or Ising formulation whose minimum solves the initial problem

Solving $A\mathbf{x} = \boldsymbol{b}$ where $A \in \mathcal{M}_n(\mathbf{R})$ and $\boldsymbol{b} \in \mathbb{R}^n$

$$i \in \{0, \cdots, n-1\}, \qquad \boldsymbol{A}_i.\mathbf{x} - b_i = 0$$

If we define the cost function as the least square distance to 0:

$$\mathcal{C}(\mathbf{x}) = \sum_{i=0}^{n-1} |\boldsymbol{A}_i.\mathbf{x} - b_i|^2 \tag{7}$$

$\mathcal{C}(\mathbf{x}) \geq 0$ per definition and $\mathcal{C}(\mathbf{x}) = 0 \iff \mathbf{x}$ solution of $A\mathbf{x} = \boldsymbol{b}$

Least square to QUBO, integer representation

N.B.: it is possible to solve non integer systems, but for benchmark, integers are fine

Least square to QUBO, integer representation

N.B.: it is possible to solve non integer systems, but for benchmark, integers are fine

SotA: Integer representation for least square QUBO formulation

$$\mathbf{x}_i = \sum_{j=0}^{r-1} x_{ij} 2^j - x_{ir} 2^r \tag{8}$$

Allows to express any integer in $\{-2^{r-1}, \cdots, 2^{r-1} - 1\}$ (same as 2's complement)

Least square to QUBO, integer representation

N.B.: it is possible to solve non integer systems, but for benchmark, integers are fine

SotA: Integer representation for least square QUBO formulation

$$\mathbf{x}_i = \sum_{j=0}^{r-1} x_{ij} 2^j - x_{ir} 2^r \tag{8}$$

Allows to express any integer in $\{-2^{r-1}, \cdots, 2^{r-1} - 1\}$ (same as 2's complement)

$$c_i(\mathbf{x}) = \left[\sum_{j=0}^{N-1} a_{ij} \left(\sum_{k=0}^{r-1} x_{jk} 2^k - x_{jr} 2^r \right) - b_i \right]^2 \quad \text{and} \quad \mathcal{C}(\mathbf{x}) = \sum_{i=0}^{n-1} c_i(\mathbf{x}) \tag{9}$$

$\mathcal{C}(\mathbf{x})$ is quadratic, binary, its minimum is what we want$=$ a QUBO problem

First results and caveat of this method

The higher order coefficients of the QUBO cost function: $\mathcal{M}(\boldsymbol{x}) \propto 2^{2r} \sum_{i=0}^{n-1} x_{i,r}$

First results and caveat of this method

The higher order coefficients of the QUBO cost function: $\mathcal{M}(\boldsymbol{x}) \propto 2^{2r} \sum_{i=0}^{n-1} x_{i,r}$

Non scalability issue

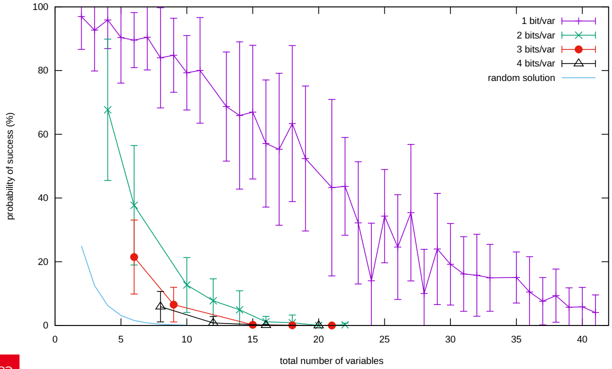
When the resolution r rises, the coefficients of the QUBO problem (or equiv. the Ising Hamiltonian) increase exponentially

First results and caveat of this method

The higher order coefficients of the QUBO cost function: $\mathcal{M}(\boldsymbol{x}) \propto 2^{2r} \sum_{i=0}^{n-1} x_{i,r}$

Non scalability issue

When the resolution r rises, the coefficients of the QUBO problem (or equiv. the Ising Hamiltonian) increase exponentially



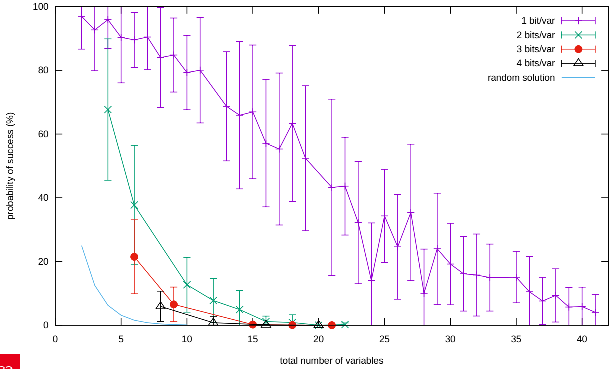
Results on DW-2000Q
(the resolution on the couplers' setting is limited to about 10^{-3})

First results and caveat of this method

The higher order coefficients of the QUBO cost function: $\mathcal{M}(\boldsymbol{x}) \propto 2^{2r} \sum_{i=0}^{n-1} x_{i,r}$

Non scalability issue

When the resolution r rises, the coefficients of the QUBO problem (or equiv. the Ising Hamiltonian) increase exponentially



Results on DW-2000Q
(the resolution on the couplers' setting is limited to about 10^{-3})

The method is not scalable

First simple benchmark: 1 bit resolution, LSS

Limiting to binary solutions (1 bit resolution) avoids the exponential problem

- Best case scenario (albeit unrealistic)

First simple benchmark: 1 bit resolution, LSS

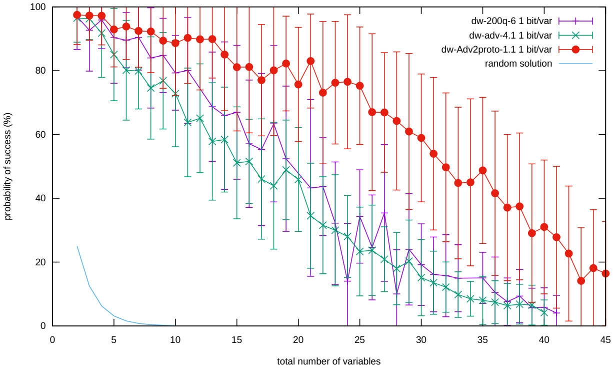
Limiting to binary solutions (1 bit resolution) avoids the exponential problem

- Best case scenario (albeit unrealistic)
- Allows to compare generations of DWave's machines

First simple benchmark: 1 bit resolution, LSS

Limiting to binary solutions (1 bit resolution) avoids the exponential problem

- Best case scenario (albeit unrealistic)
- Allows to compare generations of DWave's machines



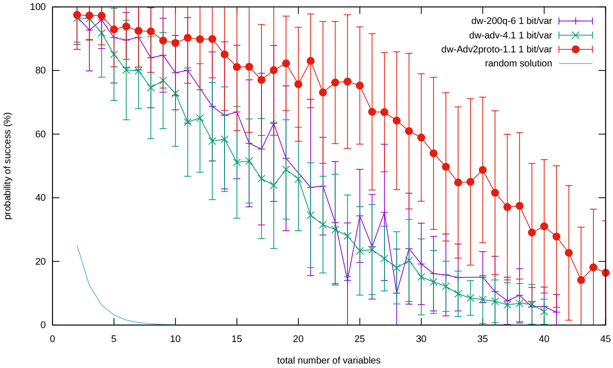
And we can fit to a sigmoid

$$\Sigma(x) = \frac{1}{1+e^{-\lambda(x-x_0)}}$$

First simple benchmark: 1 bit resolution, LSS

Limiting to binary solutions (1 bit resolution) avoids the exponential problem

- Best case scenario (albeit unrealistic)
- Allows to compare generations of DWave's machines



And we can fit to a sigmoid

$$\Sigma(x) = \frac{1}{1+e^{-\lambda(x-x_0)}}$$

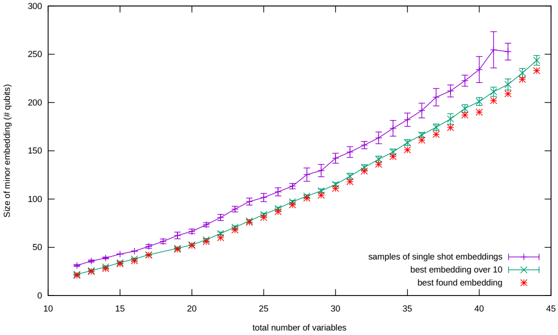
QPU generation	x_o
Chimera (dw-2000q)	20.1 ± 1.8
Pegasus (dw-advantage)	17.8 ± 0.8
Zephyr (advantage2-proto)	31.8 ± 0.8

Improving the results: tweaking the minor-embedding

Several parameters impact the quality of the results: focus on the minor-embedding problem

Improving the results: tweaking the minor-embedding

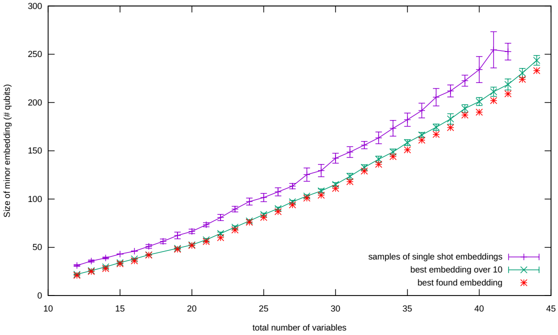
Several parameters impact the quality of the results: focus on the minor-embedding problem

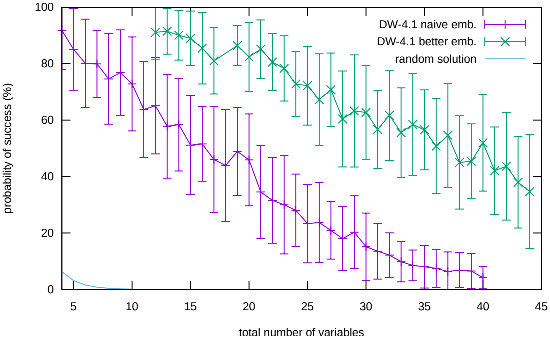


Idea= running several time the embedding heuristic and select the best

Improving the results: tweaking the minor-embedding

Several parameters impact the quality of the results: focus on the minor-embedding problem





Idea= running several time the embedding heuristic and select the best

inflection point: $17.8 \pm 0.8 \longrightarrow 37.0 \pm 1.2$

Much better benchmark result

Avoiding the exponential problem, a novel algorithm

Introduction on a simple example:

$$\begin{cases} X + Y + 1 = 0 \\ -X + Y + 3 = 0 \end{cases}$$

Avoiding the exponential problem, a novel algorithm

Introduction on a simple example:

$$\begin{cases} X + Y + 1 = 0 \\ -X + Y + 3 = 0 \end{cases}$$

Using 2's complement notation this time:

Avoiding the exponential problem, a novel algorithm

Introduction on a simple example:

$$\begin{cases} X + Y + 1 = 0 \\ -X + Y + 3 = 0 \end{cases}$$

Using 2's complement notation this time: and new variables = carries

$$\left\{ \begin{array}{llllllll}
x_0 & & & +y_0 & & & +1 & -2c_0 = 0 \\
& x_1 & & & +y_1 & & & +c_0 - 2c_1 = 0 \\
& & x_2 & & & +y_2 & & +c_1 - 2c_2 = 0 \\
\hline
x_0 & & & +y_0 & & & +1 & -2c'_0 = 0 \\
x_0 & +x_1 & & & +y_1 & & +1 & +c'_0 - 2c'_1 - 4c'_2 = 0 \\
x_0 & +x_1 & +x_2 & & & +y_2 & & +c'_1 - 2c'_3 - 4c'_4 = 0
\end{array} \right. \tag{10}$$

Avoiding the exponential problem, a novel algorithm

Introduction on a simple example:

$$\begin{cases} X + Y + 1 = 0 \\ -X + Y + 3 = 0 \end{cases}$$

Using 2's complement notation this time: and new variables = carries

$$\left\{ \begin{array}{lllllllr} x_0 & & & +y_0 & & & +1 & -2c_0 = 0 \\ & x_1 & & & +y_1 & & & +c_0 - 2c_1 = 0 \\ & & x_2 & & & +y_2 & & +c_1 - 2c_2 = 0 \\ \hline x_0 & & & +y_0 & & & +1 & -2c'_0 = 0 \\ x_0 & +x_1 & & & +y_1 & & +1 & +c'_0 - 2c'_1 - 4c'_2 = 0 \\ x_0 & +x_1 & +x_2 & & & +y_2 & & +c'_1 - 2c'_3 - 4c'_4 = 0 \end{array} \right. \tag{10}$$

And the cost function (QUBO problem):

$$\mathcal{C}(X, Y) = (x_0 + y_0 + 1 - 2c_0)^2 + \cdots + (x_0 + x_1 + x_2 + y_2 + c'_1 - 2c'_3 - 4c'_4)^2$$

Pro and cons

Pro:

- No exponential explosion of QUBO coefficients

Pro and cons

Pro:

- No exponential explosion of QUBO coefficients
- Transforming the system of equation into QUBO is only $\mathcal{O}(n^2)$

Pro and cons

Pro:

- No exponential explosion of QUBO coefficients
- Transforming the system of equation into QUBO is only $\mathcal{O}(n^2)$
- Can hint at a potential polynomial advantage of QA (with a lot of unrealistic hypotheses)

Pro and cons

Pro:

- No exponential explosion of QUBO coefficients
- Transforming the system of equation into QUBO is only $\mathcal{O}(n^2)$
- Can hint at a potential polynomial advantage of QA (with a lot of unrealistic hypotheses)
- It is straightforward to check that a solution is found

Cons:

- A sizable (albeit not unmanageable) number of auxiliary variables (the carries)
- The spectral gap of the target Hamiltonian is generally low (there are workarounds, e.g. larger annealing time)
- Avoiding spurious solutions requires to normalize the a_{ij} coefficients per line and check all the x_{ik} appear in the cost function

Pro and cons

Pro:

- No exponential explosion of QUBO coefficients
- Transforming the system of equation into QUBO is only $\mathcal{O}(n^2)$
- Can hint at a potential polynomial advantage of QA (with a lot of unrealistic hypotheses)
- It is straightforward to check that a solution is found

Cons:

- A sizable (albeit not unmanageable) number of auxiliary variables (the carries)
- The spectral gap of the target Hamiltonian is generally low (there are workarounds, e.g. larger annealing time)
- Avoiding spurious solutions requires to normalize the a_{ij} coefficients per line and check all the x_{ik} appear in the cost function

Nonetheless, this provide the base of an utility measure of solving linear systems with QA. We choose a large probability of finding the solution e.g. 20%

Experimental results

	2 vars, 3bits/v		2 vars, 4bits/v		3 vars, 3bits/v		3 vars, 4bits/v	
	std	2's	std	2's	std	2's	std	2's
Adv-4.1	22.1±6.3%	21.2±25%	4.5±3.4%	2.7±3.3%	2.0%	2.9%	0.12%	≈ 0.01%
Adv2-proto 40μs annealing	na	na	na	na	2.0%	2.7%	0.12%	0.19%
Adv2-proto 1ms annealing	25.3±21%	29.5±18%	9.4±7.9%	17.1±17%	1.9%	5.8%	0.25%	0.65%

Experimental results

	2 vars, 3bits/v		2 vars, 4bits/v		3 vars, 3bits/v		3 vars, 4bits/v	
	std	2's	std	2's	std	2's	std	2's
Adv-4.1	**22.1$\pm$6.3%**	21.2$\pm$25%	**4.5$\pm$3.4%**	2.7$\pm$3.3%	2.0%	**2.9%**	**0.12%**	$\approx$ 0.01%
Adv2-proto 40μs annealing	na	na	na	na	2.0%	**2.7%**	0.12%	**0.19%**
Adv2-proto 1ms annealing	25.3$\pm$21%	**29.5$\pm$18%**	9.4$\pm$7.9%	**17.1$\pm$17%**	1.9%	**5.8%**	0.25%	**0.65%**

Without further improvements, the results would be a utility level of 2 variables with 3 bits of resolutions for LSS on D-Wave QA

Experimental results

	2 vars, 3bits/v		2 vars, 4bits/v		3 vars, 3bits/v		3 vars, 4bits/v	
	std	2's	std	2's	std	2's	std	2's
Adv-4.1	**22.1±6.3%**	21.2±25%	**4.5±3.4%**	2.7±3.3%	2.0%	**2.9%**	**0.12%**	≈ 0.01%
Adv2-proto 40μs annealing	na	na	na	na	2.0%	**2.7%**	0.12%	**0.19%**
Adv2-proto 1ms annealing	25.3±21%	**29.5±18%**	9.4±7.9%	**17.1±17%**	1.9%	**5.8%**	0.25%	**0.65%**

Without further improvements, the results would be a utility level of 2 variables with 3 bits of resolutions for LSS on D-Wave QA

The results really point at a **spectral gap issue**

- Next urgent step: mitigate the spectral gap issue and check if it improves

Conclusion and future work

- Contrary to intuition, Quantum Annealing (and other Quantum simulators) are much more versatile than initially thought (e.g. LSS)
- It is also possible to think there can be a (polynomial) Quantum Advantage

Conclusion and future work

- Contrary to intuition, Quantum Annealing (and other Quantum simulators) are much more versatile than initially thought (e.g. LSS)
- It is also possible to think there can be a (polynomial) Quantum Advantage
- We are not there yet: but lot of possible further improvements

Conclusion and future work

- Contrary to intuition, Quantum Annealing (and other Quantum simulators) are much more versatile than initially thought (e.g. LSS)
- It is also possible to think there can be a (polynomial) Quantum Advantage
- We are not there yet: but lot of possible further improvements
- We have a path to make the whole set of Application Benchmark pieces onto D-Wave QA:
  - Many-body computing
  - LLS, Prime factorization, constrained and unconstrained optimization (including Q-Score)
  - May later include later some form of QML

Conclusion and future work

- Contrary to intuition, Quantum Annealing (and other Quantum simulators) are much more versatile than initially thought (e.g. LSS)
- It is also possible to think there can be a (polynomial) Quantum Advantage
- We are not there yet: but lot of possible further improvements
- We have a path to make the whole set of Application Benchmark pieces onto D-Wave QA:
  - Many-body computing
  - LLS, Prime factorization, constrained and unconstrained optimization (including Q-Score)
  - May later include later some form of QML
- Within 6-8 months, 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
  - We are open to discuss the relevance and improvements with tech providers or users

Conclusion and future work

- Contrary to intuition, Quantum Annealing (and other Quantum simulators) are much more versatile than initially thought (e.g. LSS)
- It is also possible to think there can be a (polynomial) Quantum Advantage
- We are not there yet: but lot of possible further improvements
- We have a path to make the whole set of Application Benchmark pieces onto D-Wave QA:
  - Many-body computing
  - LLS, Prime factorization, constrained and unconstrained optimization (including Q-Score)
  - May later include later some form of QML
- Within 6-8 months, 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
  - We are open to discuss the relevance and improvements with tech providers or users

Future work:

- Think about generic ways of improving the spectral gap in target Hamiltonian
- Porting to other QPUs:
  - Work should start in early 2026-Q1 for Pasqal's Ruby QPU
  - Same for Lumy (Quandela)
  - if possible try IBM Quantum and IQM

cea

Merci

Thanks!

CEA SACLAY
91191 Gif-sur-Yvette Cedex
France

Standard. + 33 1 69 08 60 00