A Quantum Computational Approach to Correspondence Problems on Point Sets —Supplementary Material about the Experiments on D-Wave 2000Q—

Vladislav Golyanik Christian Theobalt
Max Planck Institute for Informatics, Saarland Informatics Campus

Abstract

This supplementary material reports experimental results of Quantum Approach (QA) [4] on quantum annealer D-Wave 2000Q obtained in April 2020. We confirm that the method, which was experimentally evaluated in [4] on a simulated sampler, performs on real quantum hardware as the theory predicts. The total experimental time on a quantum processing unit in our experiments amounts to 60 *seconds which was sufficient to run over* 70 *experiments with over* $3 \cdot 10^5$ *annealings in total.*

A. Overview

We access quantum hardware remotely via *Ocean Tools Leap 2* [3], *i.e.,* a web-browser-based user interface for programming D-Wave 2000Q (Secs. B,C). D-Wave provides 60 seconds of annealing time on their machines for research purposes per month free of charge. We use this available time to confirm the effectiveness of QA experimentally on real quantum hardware in the 2D case. In total, this time suffices to perform $\sim$70 experiments in total (some experiments have been performed multiple times to confirm that the outcomes agree). A small portion of the available runtime ($\sim 5\%$) has been also spent for the 3D case.

We perform the experiment with the *fish* data set and prepare the $\mathbf{P}$ state on a CPU as described in Secs. 5 and 6 of the main matter for the transformation estimation and point set alignment, respectively. We test different initial misalignments and sizes of the point interaction regions. In a subset of the experiments, we add noise to the template resulting in several template noise ratios (24%, 35% and 43%).

B. Programming D-Wave 2000Q

For uploading $\mathbf{P}$ to the quantum annealer, we need to bring it to the upper-triangular form. Since $\mathbf{P}$ is symmetric, the equivalent upper-triangular form for QUBOP is achieved by doubling every weight above the diagonal and setting the elements below the diagonal to zero. Note that the application programming interface of D-Wave allows to fix the variables to zero or one for the entire optimisation. If a variable is fixed to zero, the equivalent form of $\mathbf{P}$ is achieved by removing the rows and columns from $\mathbf{P}$ corresponding to this variable. If a variable is fixed to one, then, additionally, the bias of the eliminated variable is added to the energy offset and the coupling weights are added to the biases of the variables with which the eliminated variable is interacting. When loading $\mathbf{P}$ on the quantum chip, the biases and couplings are scaled to enable more balanced mapping to the range of values natively supported by the hardware. At the same time, we can also uniformly scale $\mathbf{P}$ in advance so that the maximum weight in $\mathbf{P}$ does not exceed 10^2.

The QA script consists of $\sim$50 lines of *python* code, see Fig. 1 for an excerpt. We specify that our problem is defined in terms of binary variables (in contrast to the *spin mode* with -1 and 1 values) and perform automatic minor embedding of $\mathbf{P}$ to the Chimera graph [1]. Next, we set zero offset to the energy, pass $\mathbf{P}$ to the binary quadratic problem solver (saved in the variable *quadratic*) and fix the first variable to 1 (corresponding to the reference point set, since the first row of $\mathbf{\Phi}$ contains vectorised points of the reference point set which are fixed in our problems). Next, we initialise the solver and demand that the execution has to be performed entirely on a quantum processing unit. We sample $5k$ times and choose the solution for the unembedding with the lowest energy among all runs.

```
...

# the QUBOP is defined in terms of binary variables
vartype = dimod.BINARY
# vartype = dimod.SPIN

# define the upper-triangular matrix bqm which is passed to the sampler
bqm = dimod.BinaryQuadraticModel(linear, quadratic, offset, vartype)

# the first variable is fixed to 1 throughout the entire optimisation
bqm.fix_variable(1, 1)

# initialise D-WAVE 2000Q with an automatic minor-embedding
sampler = EmbeddingComposite(DWaveSampler(solver={'qpu': True}))

# run QUBOP on the QPU, repeat the sampling 5k times
sampleset = sampler.sample(bqm, num_reads=5000)

# save the result with the lowest energy among all runs
best_sample = sampleset.first.sample

# perform unembedding of the solution

...
```

Figure 1: An excerpt from our QA script.

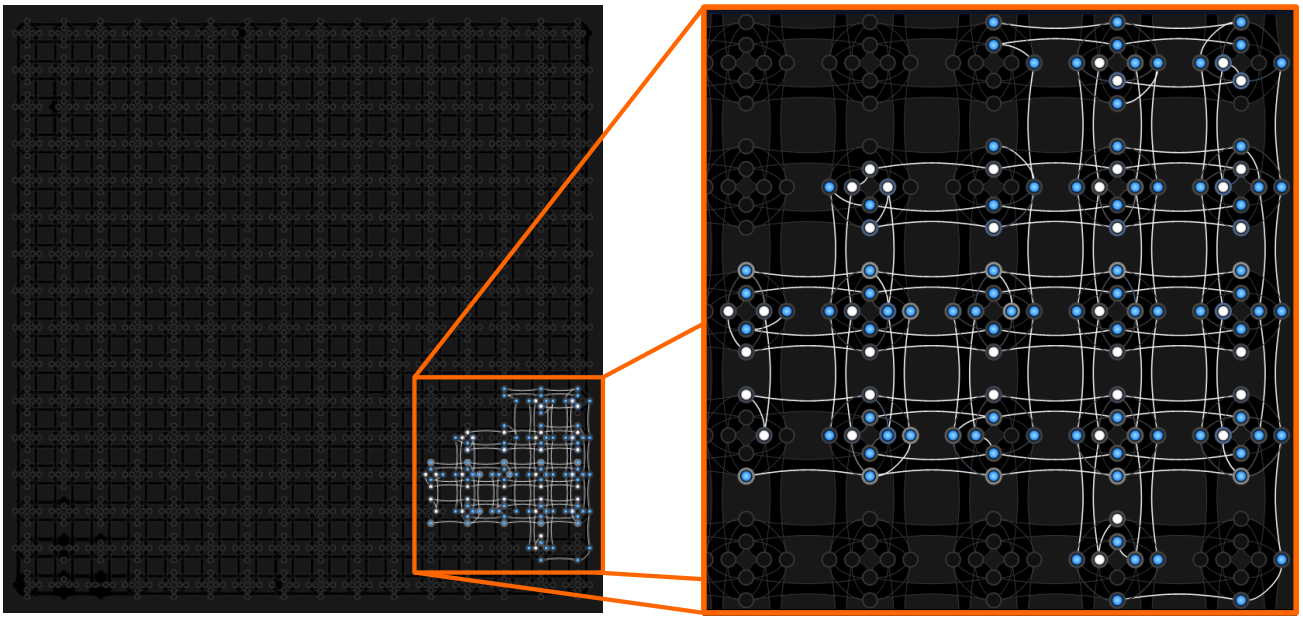

Figure 2: Exemplary minor embedding to the Chimera graph of transformation estimation in 2D. The white and blue dots denote embedded qubits which are measured as one and zero after the annealing, respectively. The problem with 20 logical qubits requires $\sim$130 physical qubits, *i.e.*, $\sim$15% of all available qubits in the Chimera graph.

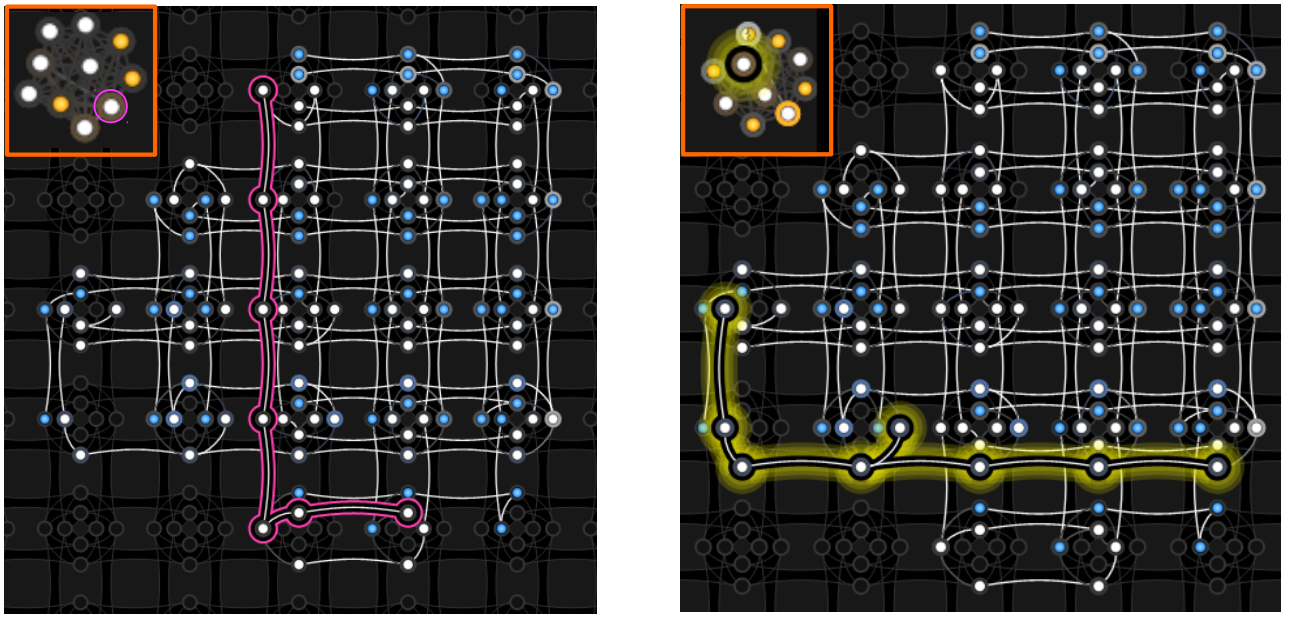

Figure 3: Chains of physical qubits (of lengths seven and eight on the left and right, respectively) in the minor embedding into the Chimera graph corresponding to a single logical qubit. The force-directed sub-views of the problem are shown in the top left corners, with the highlighted logical qubits. The programming interface warns that the chain of length eight can adversely affect the result.

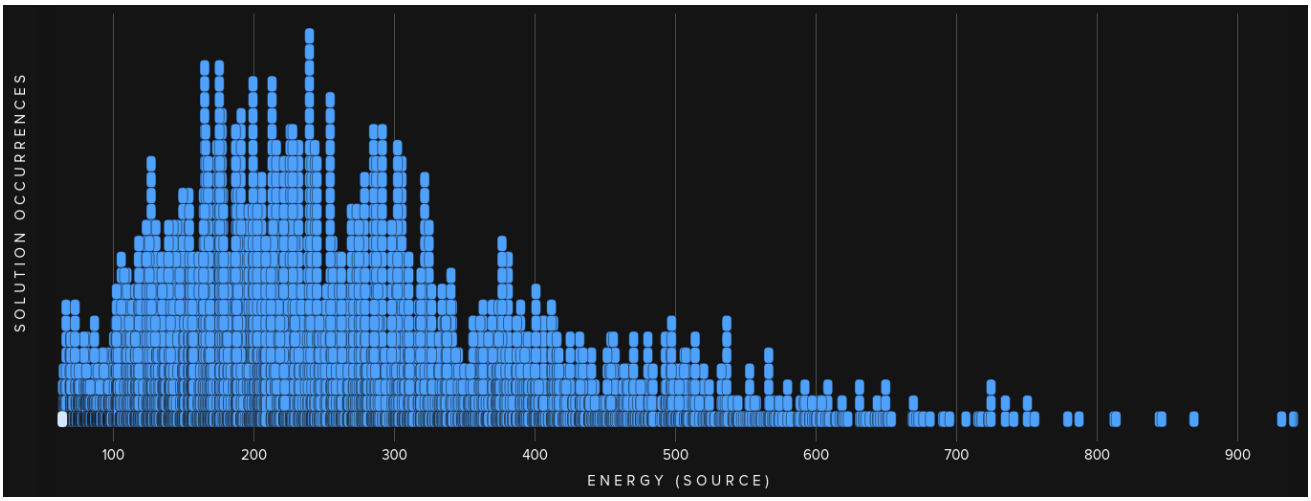

Figure 4: Solution distribution after $5k$ runs, for the transformation estimation in 2D. We take the measured bitstring for unembedding corresponding to the left-most energy (the lowest energy).

C. Minor Embedding

Fig. 2 exemplifies the result of minor embedding for transformation estimation in 2D. Mapping of our problem with 20 qubits to the Chimera architecture results in an embedding with $\sim$130 physical qubits (the exact number varies slightly for every problem instance). This suggests that multiple physical qubits are required to represent a single logical qubit in a fully

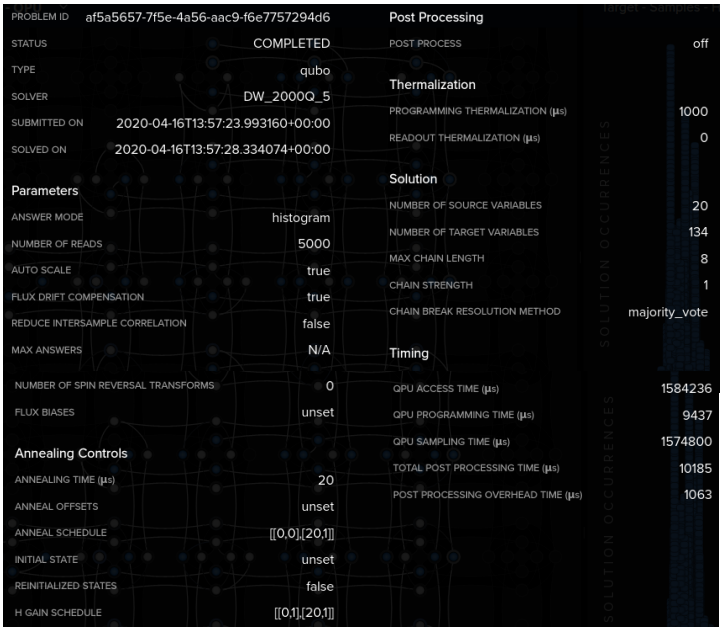

Figure 5: Problem details as displayed in the *Leap 2* D-Wave programming interface after program execution. The panel is divided into seven areas, *i.e.,* 1) problem submission information and execution status, 2) problem parameters, 3) annealing controls, 4) post-processing flag, 5) thermalisation settings, 6) information about the minor embedding ("solution") as well as 7) timing.

connected logical qubit graph. This observation agrees with the theoretical prediction, since the number of qubits within a cell which can interact with each other is limited by four. Thus, to embed logical qubit graphs with a larger number of connections, qubit copies with expansions into additional cells are necessary.

Given the structure of $\mathbf{P}$, our problem can be represented as two fully connected graphs of ten logical qubits (see Fig. 3, top-left corners). In Fig. 3, we see that up to eight physical qubits are required to represent a single logical qubit in the exemplary embedding. In the ideal case, these physical qubits—building a chain and corresponding to a single logical qubit—have to be synchronised and represent the same quantum state throughout the entire annealing. To increase the probability that the physical qubit chain does not break, *i.e.,* the measurement of all qubits in the chain results in the same value, a programmer can specify a higher chain strength instead of the default value 1.0. On the one hand, this increases the probability that the chain will not break by the end of the annealing. On the other hand, the longer the chain, the higher is the chance that it breaks and that the auxiliary constraints supporting the chain—which do not have anything to do with the actual problem formulation—will adversely affect the result (even if the chain is not broken).

The solution based on a broken chain is unreliable and should be statistically discarded (it is though probable that such solution will be optimal). The broken chain correction method of D-Wave is the majority vote of the qubit states in a chain [2] (see also Fig. 5). All in all, a high number of repetitions statistically compensates invalid results based on broken chains in practice. The analysis of an optimal number of repetitions and chain strengths under the consideration of chain lengths required by the minor embedding can be performed in future research.

D. Solution Distribution and Problem Details

In total, we sample $5k$ times and forward the *first sample*, *i.e.,* the sample with the lowest energy among all runs (the samples are ordered from the lowest energies to the highest energies), to unembedding. The histogram of resulting energies is shown in Fig. 4. Note that the most frequent occurrence does not have the lowest energy and does not correspond to the most accurate solution. While this is the case in QA, future adiabatic quantum algorithms can be also designed in such a way that the most frequent occurrence corresponds to the optimal solution (see, *e.g.,* the recent example [6]). From the solution distribution, we also can see that we would have a high chance of obtaining an optimal solution by QA if we would sample just $1k$ or 500 times.

Exemplary problem details after a successful completion of quantum annealings— corresponding to the the minor embedding in Fig. 3-(left)—are provided in Fig. 5. In the panel partitioned in seven areas, we can see, among others, the problem identifier, the execution status and the name of the quantum annealer which has run the submitted program. We can read statistics about the logical number of qubits (*number of source variables*), the number of physical qubits in the minor embedding (*number of target variables*), the maximum physical qubit chain length and the used chain strength. In this example, the number of physical qubits amounts to 134, and the maximum chain length is eight. Timing context provides low-level details about the total QPU access, programming and sampling time, as well as post-processing time. We also see the default annealing time $\tau = 20\mu s$ used in all our experiments.

Selected alignment results obtained on D-Wave for $K \in \{1, 20, 30\}$ are visualised in Fig. 6. Experimental outcomes on D-Wave agree with the results of simulated sampling, with the difference that multiple annealings have to be performed to select the optimal solution. In all experiments, we observe expected registrations with small deviations in $e_{\mathbf{R}}$ from a valid rotation in 2D, similarly as reported for the classical sampler. For instance, under the angle of initial misalignment of π and $K = 1$, we obtain $e_{\mathbf{R}} = 0.13789$ which agrees with the value reported in Fig. 3-B of the main matter.

We observe that the results are obtained on D-Wave faster than on the CPU sampler, *i.e.,* through exhaustive search over binary vectors. This experiment also confirms that our method works as the theory predicts. *Not only an individual anneal is significantly faster that an exhaustive search on a CPU, but also the total time required to reinitialise the problem and sample it $5k$ times takes less time, at least for the largest tested problems*.

E. A Remark on the 3D Case

We have also performed tests with the point cloud *galleon* [5] in 3D (transformation estimation). Due to a high number of logical fully connected qubits due to the larger number of elements in the basis (70), the embedding with 70 qubits is not successful. With a reduced basis and 30 logical qubits, the embedding is possible, however, the annealing has not resulted in optimal alignment. We have also tested $\mathbf{P}$ with a complete basis on a hybrid solver, *i.e.,* operational mode which starts with a set of proprietary heuristics and finishes with an annealing on a QPU. We obtain an alignment which closer meets theoretical predictions. Fig. 7-(A) shows the 3D example which we have tested on a hybrid solver, *i.e.,* initial misalignment and the registration result from two perspectives. Fig. 7-(B) visualises the minor embedding of QA for the 3D case with a reduced basis which requires 30 qubits. Compared to the 2D case with the full basis, the number of physical qubits required to map $\mathbf{P}$ grows in this case by $\sim 275\%$. Note that embedding with the full basis which requires 70 qubits cannot be accomplished on the current D-Wave generation. Unfortunately, since the code of the hybrid solver is closed-source, no reliable scientific conclusions can be made in regard to the outcome of this experiment. Future generations of quantum annealers will enable a more thorough analysis of the 3D case.

F. Discussion

We were able to execute and confirm QA on D-Wave successfully. The D-Wave compiler successfully embedded all tested 2D problems, uploaded those to the AQC (including the conversion of the weights to the local magnetic fields) and initialised the qubits in the $|+\rangle$ state. After the sampling, we have unembedded the solution with the lowest energy among all runs. The experiment confirms that our method as well as the D-Wave quantum annealer work as the theory predicts. Next, we have not observed differences in the final result with different chain strengths in 2D. The reason is that even though the distribution of the solutions changes with different chain strengths, their is still the same solution with the lowest energy present in the distribution which we unembed.

AQC opens up possibilities to solve computer vision and pattern recognition problems in new ways which are yet to be explored in the next decades. The theory states that due to quantum effects and speed-up, many combinatorial problems can be solved faster compared to algorithms for classical hardware, with obtained solutions being closer to the globally-optimal minimisers. AQC programming also brings a *paradigm shift*, as the new hardware requires new algorithm design methodology. On the one hand, new mathematical formulations will be required to map a problem—possibly with constraints—to QUBOP. On the other hand, existing formulations which would lead to overheads and unnecessarily complicate a classical algorithm, can be re-considered and perhaps successfully applied in the quantum case.

Speaking about alignment problems, there is a magnitude of directions for future work. QA searches for a solution in the space of affine transformations, and the obtained solution is an approximation of a rotation. Thus, QA can be extended for affine registrations in a straightforward way. The initial formulation for point set alignment (23) allows embedding of prior correspondences and other alignment cues through point masses. Such embedding of boundary conditions can also be explored in QA. Next, there are formulations of alignment problems (on meshes and point clouds) with permutations. It

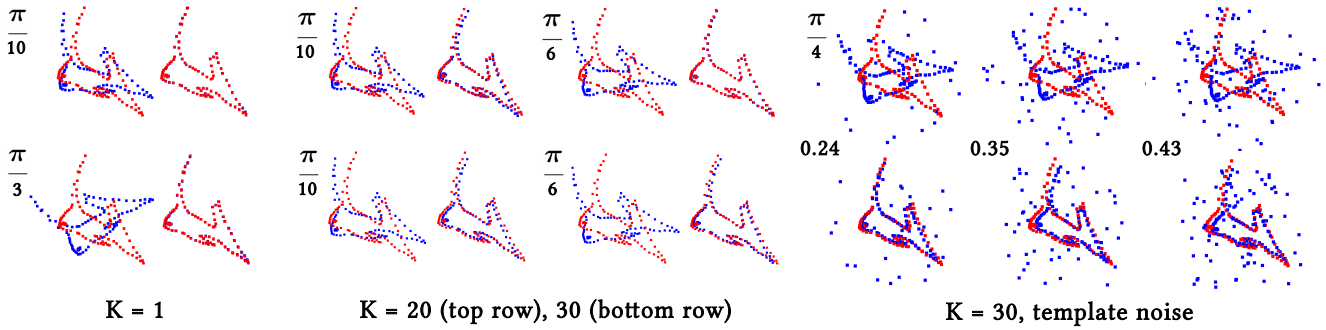

Figure 6: Point set alignment results obtained on quantum annealer D-Wave 2000Q. The angle of initial misalignment is given to the left of the initial configurations. In the experiments without added noise (on the left and in the middle), the results are shown to the right from the initialisations. In the experiments with added noise (on the right), the results are shown on the bottom from the initialisations, and the numbers 0.24, 0.35 and 0.43 denote the noise ratios. K determines the number of interacting points, for every point of the reference (the points are centred around the corresponding points between the template and reference).

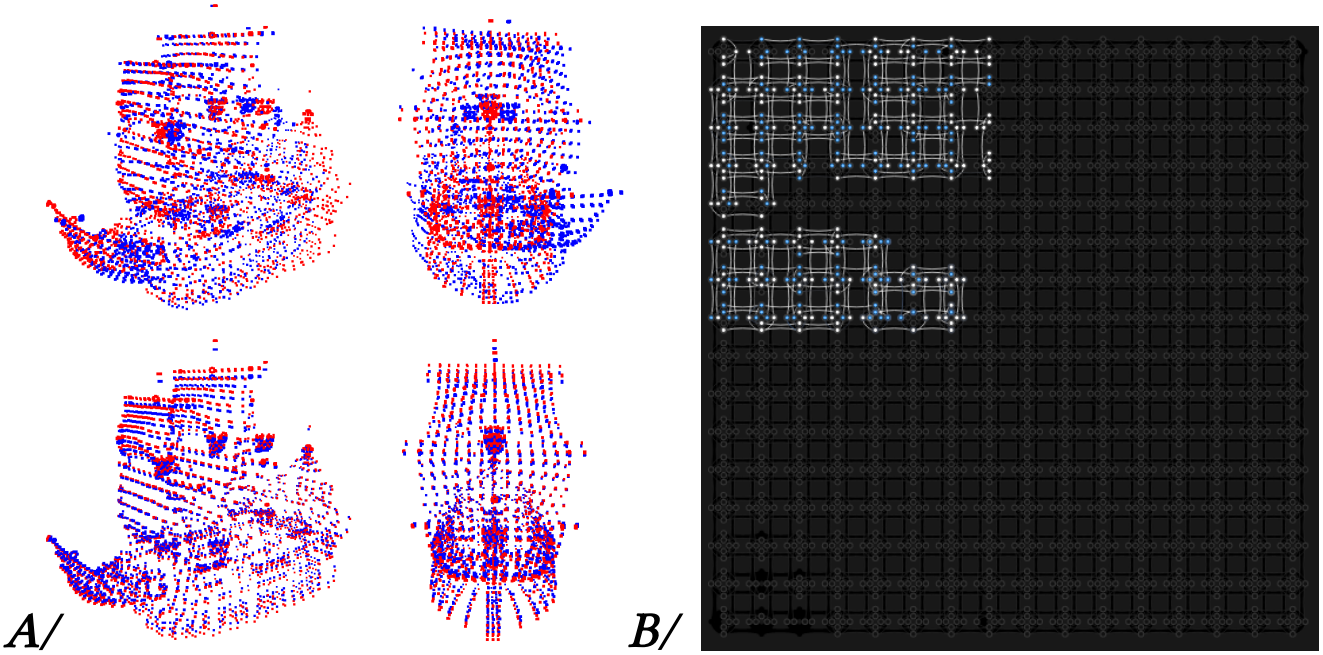

Figure 7: ***A/:*** Initialisation and alignment results on 3D *galleon* [5] with a hybrid D-Wave solver (initialisation from two perspectives on the top row and alignment results on the bottom row), ***B/:*** minor embedding for the 3D case with a reduced basis of 30 logical qubits. Compared to the 2D case (Fig. 2), the embedding requires $\sim$2.75 times more physical qubits.

is still an open question how hard permutation matrix constraints can be imposed in a QUBO for such problems as graph matching [6] or mesh registration. Another research question is how QA can be extended for the non-rigid case.

References

[1] Jun Cai, William G. Macready, and Aidan Roy. A practical heuristic for finding graph minors. *arXiv e-prints*, 2014. 1

[2] D-Wave Systems. dimod Documentation, Release 0.6.9. https://docs.ocean.dwavesys.com/_/downloads/dimod/en/0.6.9/pdf/, 2019. online; accessed 26 August 2020. 4

[3] D-Wave Systems, Inc. D-Wave Leap: Take the Leap. https://cloud.dwavesys.com/leap/, 2020. online; accessed on the 8 April 2020. 1

[4] Vladislav Golyanik and Christian Theobalt. A quantum computational approach to correspondence problems on point sets. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[5] John Burkardt. John Burkardt's Home Page. https://people.sc.fsu.edu/~jburkardt/data/ply/ply.html, 2020. online; accessed on the 8 April 2020. 5, 6

[6] Marcel Seelbach Benkner, Vladislav Golyanik, Christian Theobalt, and Michael Moeller. Adiabatic quantum graph matching with permutation matrix constraints. In *International Conference on 3D Vision (3DV)*, 2020. 4, 6