Quantum-Classical Solution Methods for Binary Compressive Sensing Problems

Robert S. Wezeman[1], Irina Chiscop[1], Laura Anitori[2], and Wim van Rossum[2]

The Netherlands Organisation for Applied Scientific Research
[1] Department of Cyber Security & Robustness
[2] Department of Radar Technology
{robert.wezeman, irina.chiscop, laura.anitori, wim.vanrossum}@tno.nl

Abstract. Compressive sensing is a signal processing technique used to acquire and reconstruct sparse signals using significantly fewer measurement samples. Compressive sensing requires finding the most sparse solution to an underdetermined linear system, which is an NP-hard problem and as a consequence in practise is only solved approximately. In our work we restrict ourselves to the compressive sensing problem for the case of binary signals. For that case we have defined an equivalent formulation in terms of a quadratic binary optimisation (QUBO) problem, which we solve using classical and (hybrid--)quantum computing solving techniques based on quantum annealing. Phase transition diagrams show that this approach significantly improves the number of problem types that can be successfully reconstructed when compared to a more conventional $\mathcal{L}_1$ optimisation method. A challenge that remain is how to select optimal penalty parameters in the QUBO formulation as was shown can heavily impact the quality of the solution.

Keywords: Binary compressive sensing · Quadratic unconstrained binary optimisation · Quantum annealing

1 Introduction

The Nyquist sampling theorem states that to be able to perfectly reconstruct a signal, it has to be sampled at a rate that is at least twice the highest frequency in the original signal, the *Nyquist rate* [28]. Unfortunately for many applications, this Nyquist rate is too high for sampling to be feasible in practise. Let $\boldsymbol{x} \in \mathbb{R}^N$ be a real N-dimensional signal and $\{\boldsymbol{\Psi_i}\}_{i=1}^N$ be a set of basis vectors. We can write a signal as $\boldsymbol{x} = \sum_{i=1}^N s_i \boldsymbol{\Psi_i}$. The signal is said to be K-sparse in the basis Ψ if the N-dimensional vector $\boldsymbol{s}$ with coefficients s_i has at most K nonzero elements. Compressive sensing (CS) is a signal processing technique that makes it possible to acquire and reconstruct a signal more efficiently, given that the signal has a sparse representation ($K \ll N$) in some basis [10]. CS has a large number of applications in several fields ranging from (medical) imaging, communication systems and pattern recognition, to speech and sound processing [26].

The task of compressive sensing is equivalent to finding the most sparse solution of an underdetermined linear system. Consider that problem, recovering

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

the most sparse vector $\boldsymbol{x} \in \mathbb{R}^N$ from a undersampled set of $M \ll N$ measurements $\boldsymbol{y} \in \mathbb{R}^M$. These vectors are related by what is called the measurement matrix $A \in \mathbb{R}^{M \times N}$ which in this article has its elements drawn independent and identically distributed random from some distribution on $\mathbb{R}$. The reconstruction problem is then described as follows

$$\min_x \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad A\boldsymbol{x} = \boldsymbol{y}, \tag{1}$$

where $\|\cdot\|_0$ is the $\mathcal{L}_0$-norm, the number of non-zero elements of a vector. To guarantee a unique solution to this problem, additional constraints on the measurement matrix are required, such as the Restricted Isometry Property (RIP) [15,16]. Given that this problem can be solved uniquely, compressive sensing then enables a much shorter signal acquisition time together with reduced amounts of data, because the needed number of samples M can be much smaller than what would be required according to the Nyquist rate. Unfortunately, this problem does not have a closed form solution and furthermore, it is an NP-complete problem and thus hard to solve having combinatorial complexity [7,10].

Summarised, compressed sensing based signal acquisition allows a significant decrease in the sampling rate of sparse signals, but in return requires a hard optimisation problem to be solved. Algorithms used to solve the sparse reconstruction problem tend to be very slow and rely heavily on the speed at which matrix-vector multiplications can be done for the measurement matrix A. An overview of different computational techniques for solving the sparse reconstruction problem is given in [30]. One of the major algorithmic approaches relaxes the original problem by replacing the combinatorial $\mathcal{L}_0$ norm on $\boldsymbol{x}$ by the $\mathcal{L}_1$-norm, the sum of the magnitudes of a vector $\boldsymbol{x}$. The obtained problem, given by Eq. (2), becomes a convex optimisation problem which can then be solved using standard convex optimisation routines.

$$\min_x \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad A\boldsymbol{x} = \boldsymbol{y}. \tag{2}$$

Candès and Romberg proof that $\mathcal{L}_1$ optimisation obtains the exact solution to Eq. (1) under the constraint that the number of samples $M > C\mu^2(A)K \log N$, where $\mu(A) = \sqrt{N} \max_{k,j} |A_{k,j}|$, K is the sparsity of $\boldsymbol{x}$ and C is some constant [14]. The required computation time, even in the case when convex optimisation routines are used, can still remain an obstacle for real-time applications and is limited by the available computational resources. This is due to the iterative nature of the algorithms, in which each iteration is closely related to the corresponding processing step in conventional processing. Therefore, when many iterations are required this results in a significant increase in computations.

Meanwhile, a new computing paradigm, quantum computing, is quickly approaching us. It is expected that quantum computers will be able to solve specific problems faster than the current generation of classical computer are capable of. In [11] thirteen applications for radar and sonar information processing have been identified, among which compressive sensing, that can possibly be improved using quantum computing.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

In this work, we make first steps towards a quantum computing approach for compressive sensing. We simplify the problem by only considering binary valued signals and formulate that problem as a quadratic unconstrained binary optimisation (QUBO) problem. Binary sparse signal recovery is relevant for applications such as event detection in wireless sensor networks [29], group testing [27] and spectrum hole detection for cognitive radios [21]. Using quantum annealing techniques to solve binary CS has been studied before by Ayanzadeh et al. [8,9], however, only limited number of numerical results have there been given. We extend upon their work by solving the binary CS problem using both classical techniques as well as annealing based (hybrid-)quantum computing techniques. The obtained results are compared with a more conventional $\mathcal{L}_1$ optimisation approach by looking at phase transition diagrams.

The structure of this paper is as follows. In Section 2, we describe the considered problem of binary compressive sensing and formulate it as a QUBO problem. In Section 3, we describe solution methods and we describe how to create phase transition diagrams which can be used to quantitative compare different solution methods. In Section 4, the numerical results are described and we end with Section 5, where conclusions are given together with directions for further research.

2 Problem formulation

In this paper we focus on the original problem of recovering binary signals from a limited number of measurements, which can be formulated as follows:

$$\min_{x\in\{0,1\}} \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \|A\boldsymbol{x}-\boldsymbol{y}\|_2 = 0. \tag{3}$$

This problem is known to be NP-complete [9], however it can be tackled using quantum annealers. This type of quantum devices require either an Ising or Quadratic Unconstrained Binary Optimisation (QUBO) problem formulation [5]. As shown in [9] Eq. (3) can be translated to the following QUBO:

$$\min_{\boldsymbol{x}\in\{0,1\}^N} \gamma\|\boldsymbol{x}\|_0 + \|A\boldsymbol{x}-\boldsymbol{y}\|_2^2. \tag{4}$$

After expanding the norms, this translates to:

$$\min_{\boldsymbol{x}\in\{0,1\}^N} \quad \sum_i \Big(\gamma + \sum_l A_{li}[-2y_l + A_{li}]\Big)x_i + \sum_{i,j;i<j} \Big(2\sum_k A_{ki}A_{kj}\Big)x_i x_j. \tag{5}$$

We note that the objective in Eq. (5) represents the same optimization problem as in Eq. (4) but its optimal objective value is smaller by $\|\boldsymbol{y}\|_2^2$, since this constant term does not get taken into account in the QUBO. This formulation, which applies a penalty parameter γ to the $\mathcal{L}_0$ norm, is the only one we encountered in literature [9]. However, one can also opt to multiply a penalty parameter λ with the constraint term:

$$\min_{\boldsymbol{x}\in\{0,1\}^N} \|\boldsymbol{x}\|_0 + \lambda\|A\boldsymbol{x}-\boldsymbol{y}\|_2^2. \tag{6}$$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

which is then equivalent to:

$$\min_{\boldsymbol{x}\in\{0,1\}^N} \quad \sum_i \Big(1+\lambda\sum_l A_{li}[-2y_l+A_{li}]\Big)x_i + \sum_{i,j;i<j}\Big(2\lambda\sum_k A_{ki}A_{kj}\Big)x_ix_j. \quad (7)$$

From a theoretical point of view, the two formulations in Eq. (5) and Eq. (7) are equivalent when $\gamma = \frac{1}{\lambda}$. When γ and λ are set to appropriate values, both should lead to the optimal value of $\boldsymbol{x}$. However, quantum annealing-based solvers are sensitive to small variations in the QUBO values, due to the finite precision available [4]. This means that depending on the problem size (number of measurements M and size of signal N), and on the entries of matrix A, one formulation may results in better solution quality than the other. From now on we identify the formulation in Eq. (5) as QUBO type 1 and the formulation in Eq. (7) as QUBO type 2.

3 Solution methods and evaluation

For solving the binary CS problem we consider a classical approach for $\mathcal{L}_1$-minimisation and several quantum-based and hybrid solution methods which address $\mathcal{L}_0$-minimisation. When solving usual combinatorial problems, one would look at the optimality gap or lower and upper bounds as metrics to describe the quality of the solutions found. However, in compressive sensing, phase transition diagrams [18] are widely used to indicate whether a certain algorithm is successful in recovering the signal under certain conditions. In this section we describe all the solution methods used and elaborate on the use of phase diagrams in CS applications.

3.1 $\mathcal{L}_1$-minimisation

As already described in Section 1, signal recovery can also be tackled by solving a convex $\mathcal{L}_1$ minimisation problem, which is equivalent under certain strict conditions on the number of measurements taken. Alternating Direction Method of Multipliers (ADMM) is a technique that has proven itself to be suitable for solving distributed convex optimisation problems and in particular large-scale problems arising in compressive sensing [13]. In this paper we focus on a set of publicly available solvers [32] implemented in Matlab that apply ADMM [31] to solve the following problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \|\boldsymbol{x}\|_1 + (1/2\rho)\|A\boldsymbol{x}-\boldsymbol{y}\|_2^2. \quad (8)$$

where $\rho = \frac{0.01}{\|\boldsymbol{y}\|_\infty}$ and $\|\cdot\|_\infty$ is the infinity norm, the element with the largest absolute value of a vector. We have chosen this solver as a benchmark for $\mathcal{L}_1$-based approaches.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

3.2 $\mathcal{L}_0$-minimisation

The QUBO formulations given by Eq. (5) and Eq. (7) corresponding to the $\mathcal{L}_0$-minimisation binary CS problem, were solved using a classical commercial solver and built-in algorithms of the Ocean tool suite provided by D-Wave Systems [1]. In the following paragraphs, we provide some background on each of these algorithms.

Gurobi Optimiser is a commercial optimisation software library for solving mixed-integer linear and quadratic optimisation problems[1]. It is widely used for large-scale applications in different industries [6]. We have implemented this solver using the *gurobipy* Python package. To ensure a reasonable computation time, we have limited the Gurobi running time per problem to 30s.

Simulated Annealing (SA) is a stochastic meta-heuristic [17], which emerged as a local search method that can escape from being trapped in local optima. There are two stochastic steps in simulated annealing. First, a solution s' is chosen based from the set of neighbors of the current solution s according to some given distribution (each neighbor usually has the same probability). Then, the chosen solution is accepted with probability $p(s, s', c) = e^{\min\{c(s)-c(s'),0\}/c}$, where c is a positive control parameter which decreases with increasing number of iterations and converges to 0. The performance of the algorithm relies on the cooling schedule, i.e., which specifies the initial and final values of the control parameter c together with a decrement function. In our numerical experiments, we employed the the simulated annealing algorithm provided by D-Wave Systems with default parameters. Further details concerning the annealing (cooling) schedule and parameters can be found in the official software documentation [2]. Similar to Gurobi, we have imposed a limit of 30s to the simulated annealing process on each problem.

Quantum Annealing is a meta-heuristic which utilizes quantum fluctuations [22] and quantum computation by adiabatic evolution [20] in solving a particular type of optimisation problems, and is currently implemented in the quantum devices produced by D-Wave Systems. The evolution of a quantum state on D-Wave's Quantum Processing Unit (QPU) is described by a time-dependent Hamiltonian, composed of initial Hamiltonian H_0, whose ground state is easy to create, and final Hamiltonian H_1, whose ground state encodes the solution of the problem at hand:

$$H(t) = \Big(1 - \frac{t}{T}\Big)H_0 + \frac{t}{T}H_1. \tag{9}$$

The system in Eq. (9) is initialized in the ground state of the initial Hamiltonian, i.e. $H(0) = H_0$. The adiabatic theorem states that if the system evolves according to the Schrödinger equation, and the minimum spectral gap of $H(t)$ is not

[1] https://www.gurobi.com/products/gurobi-optimizer/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

zero, then for time T large enough, $H(T)$ will converge to the ground state of H_1, which encodes the solution of the problem. The D-Wave quantum annealer accepts as input H_1 either as an Ising Hamiltonian, or as its equivalent formulation, the QUBO. We remark that although Eq. (9) suggests that the annealing is performed linearly, in practice D-Wave uses a non-linear annealing schedule. Additionally, the chosen formulation needs to be embedded on the hardware. The current architecture of D-Wave's Advantage 4.1 quantum computing system contains approximately 5000 qubits, with a total number of 35.000 qubit couplers (each qubit is connected to 15 other qubits) [24]. When the problem size is too large, such an embedding is impossible and so the quantum annealing process cannot be applied to the entire problem. In this scenario, hybrid algorithms need to be employed, in which the problem is first decomposed into smaller sub-problems, the each of these being solved with quantum annealing and at the end, the complete solution vector is reconstructed from all sub-sample solutions. In this paper, we use the term 'quantum annealing' only for situations in which the entire problem can be passed on to the D-Wave QPU, without any need for decomposition.

QBsolv is a hybrid solver introduced by D-Wave to tackle very large optimisation problems [12]. It combines a problem decomposition step with classical search algorithms and quantum annealing to find a global optimal solution. At first, an initial solution for the entire problem is obtained using classical Tabu search. Based on this solution, the problem is split into sub-problems of smaller size by ordering the problem variables according to their impact on the objective function. Then, each of these sub-problems is solved using quantum annealing, and the current solution is updated with the corresponding bits from the sub-problem vector. Then, a new Tabu search is applied starting from the current solution, and the algorithm loop is repeated until a maximum number of iterations is reached. QBsolv makes use of the different numerical precision in classical algorithms and quantum annealing. Whilst solving the sub-problems with quantum annealing on D-Wave is affected by limited precision, the classical Tabu search can solve the full problem QUBO in the standard format of IEEE double-precision (64-bit) floating-point values [12]. In this paper we have used the built-in implementation of the D-Wave Ocean tool suite[2] with default parameters.

Hybrid Solvers are offered by D-Wave Systems to enable solving arbitrarily large optimisation problems [3]. There are two types of resources available: cloud-based hybrid solvers (also known as Leap hybrid solvers) and the *dwave-hybrid* Python framework, which allows the creation of custom hybrid workflows. We focused on the first category, since according to D-Wave's software documentation [3], these solvers implement state-of-the-art classical algorithms together with intelligent allocation of the QPU to sections of the problem where it is most beneficial. The classical components herein utilize quantum queries to the D-

[2] https://github.com/dwavesystems/qbsolv/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

Wave QPU to guide their search of the larger solution space. Whilst the generic structure of Leap's hybrid solvers is described in a technical report [25], no details concerning the implemented algorithms (which classical meta-heuristics are implemented, how they utilize the solution output from the QPU etc.) are disclosed. We therefore remain cautious when interpreting the results obtained with the LeapHybridSampler, the hybrid solver that was selected for our problem.

3.3 Phase transition diagrams

Maleki and Donoho [23] have developed a framework that can be used to compare quantitatively the properties of different reconstruction algorithms. Performance is measured by what is called the undersampling-sparsity trade-off, and can be visualised by a phase transition curve. The idea is to find the "breakdown point", the point where an algorithm can still successfully reconstruct a sparse solution provided K is smaller than a certain definite fraction of N.

We define $\delta = \frac{M}{N}$ to be a measure of problem indeterminacy and define $\rho = \frac{K}{M}$ to be a measure for the sparsity of the problem. The difficulty of a problem instance can be visualised by its point in the two-dimensional phase space $(\delta, \rho) \in [0,1] \times [0,1]$. In Figure 1, examples are shown how a typical phase transition diagram looks, in this case for different problem sizes, using the $\mathcal{L}_1$ optimisation recovery approach as described in Section 3.1. The hardest problems are in the top left of this plane, trying to reconstruct a not so sparse signal with relatively few measurements. The different colors represent the probability for successful reconstruction of the sparsest solution. This figure illustrates the typically behaviour of reconstruction algorithms, below a certain threshold the algorithms works well, while above that threshold reconstruction fails. Typically the transition zone becomes narrow and better defined for larger problem sizes and depends on both the reconstruction algorithm but also the type of measurement matrix [23].

In this paper we will only consider Gaussian random measurement matrices. A Gaussian random measurement matrix can be generated by sampling its elements a_{ij} independent and identically distributed from the standard normal distribution with mean zero and standard deviation one. This type of matrix is commonly used in CS as it is known to satisfy the RIP condition with high probability [15] and has its phase transition shape well studied [19].

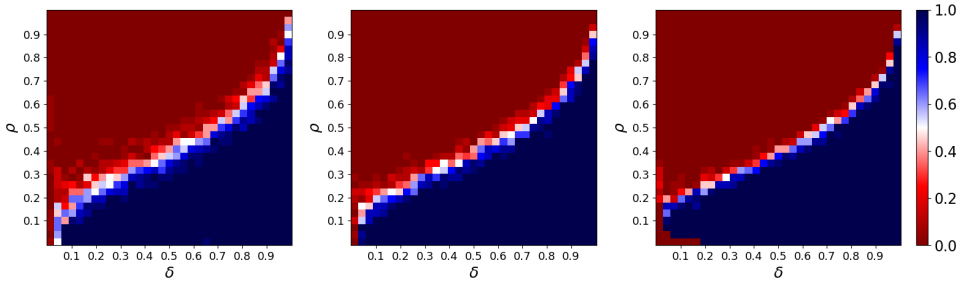
For a given reconstruction algorithm, for a fixed problem size N, the construction of phase transition diagrams is as follows:

1. Create a grid by varying both parameters δ and ρ in $(0, 1)$.
2. For each combination (δ, ρ) we calculate $M = \lceil \delta N \rceil$ and $K = \lceil \delta \rho N \rceil$.
3. At each combination (δ, ρ) we create L problem instances $(A, \boldsymbol{x}, \boldsymbol{y})$ and obtain L algorithm outputs $\boldsymbol{x}_{solve}$.
4. For each problem instance we declare it as a successful reconstruction if

$$\frac{\|\boldsymbol{x} - \boldsymbol{x}_{solve}\|_2}{\|\boldsymbol{x}\|_2} \leq tol,$$

for some tolerance parameter tol, where $\|\cdot\|_2$ is the standard Euclidean norm.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9



(a) $N = 150$ (b) $N = 300$ (c) $N = 600$

Fig. 1: Phase transition plots generated with Matlab-$\mathcal{L}_1$ solver for Gaussian random measurement matrices and 20 problem instances per step. The color indicate the probability that reconstruction for $(\delta = \frac{M}{N}, \rho = \frac{K}{M})$ instances is successful.

4 Results

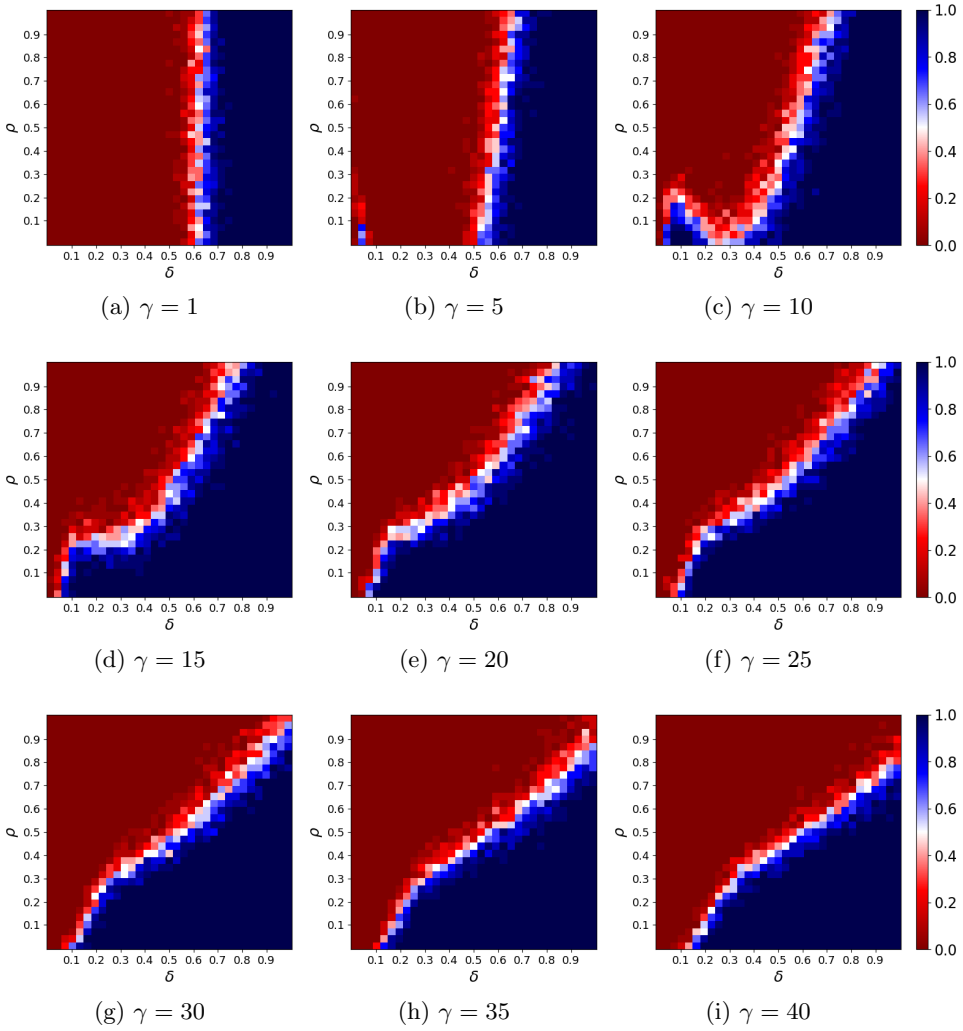
In this section we present an overview of the various numerical experiments that we performed. We studied different aspects of binary compressing sensing problems focusing on two aspects: the QUBO penalty parameter (γ or λ) and the QUBO formulation. Once suitable choices have been identified for these parameters, we also performed a comparison of different classical and (hybrid-)quantum solvers. All phase transition plots in this chapter are generated with an error tolerance value $tol = 0.1$ and a grid step size of 33 for both δ and ρ.

4.1 Impact of penalty parameter

When using a QUBO-based approach to solve the binary compressive sensing, we need to find suitable values for the penalty parameter γ (or λ). To this end, we created phase diagrams for signal size $N = 300$, using simulated annealing for $\gamma \in \{1, 5, 10, 15, 20, 25, 30, 35, 40\}$. We have chosen simulated annealing due to limited computation time available on the D-Wave QPU. Figure 2 shows all the resulting diagrams. From this figure we observe that the optimal choice of γ ranges between 15 and 25, as these values result in the largest success regions. It is also worthwhile to notice that in the case of $\gamma \in \{1, 5, 10\}$ the Simulated Annealing algorithm achieves successful signal recovery for $\delta > 0.7$, independent of the sparsity of the signal. The phase transition diagrams obtained for $\gamma = 5$, or $\gamma = 10$ look counter-intuitive. In these examples, if the algorithm is successful in the $\delta, \rho \in [0, 0.1] \times [0, 0.1]$ where the signal is sparse and the measurements are very few, it should theoretically also be capable of recovering signals with the same sparsity when more measurements are done, i.e. in regions $\delta, \rho \in [0, 0.1] \times [0.1, 0.4]$. This phenomenon could be explained by the time-out limit of 30s set on the simulated annealing solver, which may struggle to distinguish between many 'good' solutions in the search space. The variation in

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

obtained phase transition suggests that γ is dependent not only on δ and ρ (or equivalently, M, N, K) but also on the entries of the matrix A. The optimal value of γ obtained using simulated annealing seems to be a suitable choice for QBsolv as well. Figure 3a shows the phase transition diagram obtained with QBsolv and $\gamma = 20$. The region of successful signal recovery is quite large, especially when compared to any of the simulated annealing phase transition diagrams.



(a) $\gamma = 1$ (b) $\gamma = 5$ (c) $\gamma = 10$

(d) $\gamma = 15$ (e) $\gamma = 20$ (f) $\gamma = 25$

(g) $\gamma = 30$ (h) $\gamma = 35$ (i) $\gamma = 40$

Fig. 2: Phase transition diagrams generated for $N = 300$ using QUBO type 1, Simulated Annealing and 20 problem instances per step. The color indicate the probability that reconstruction for $(\delta = \frac{M}{N}, \rho = \frac{K}{M})$ instances is successful.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

4.2 Evaluation of different QUBO formulations

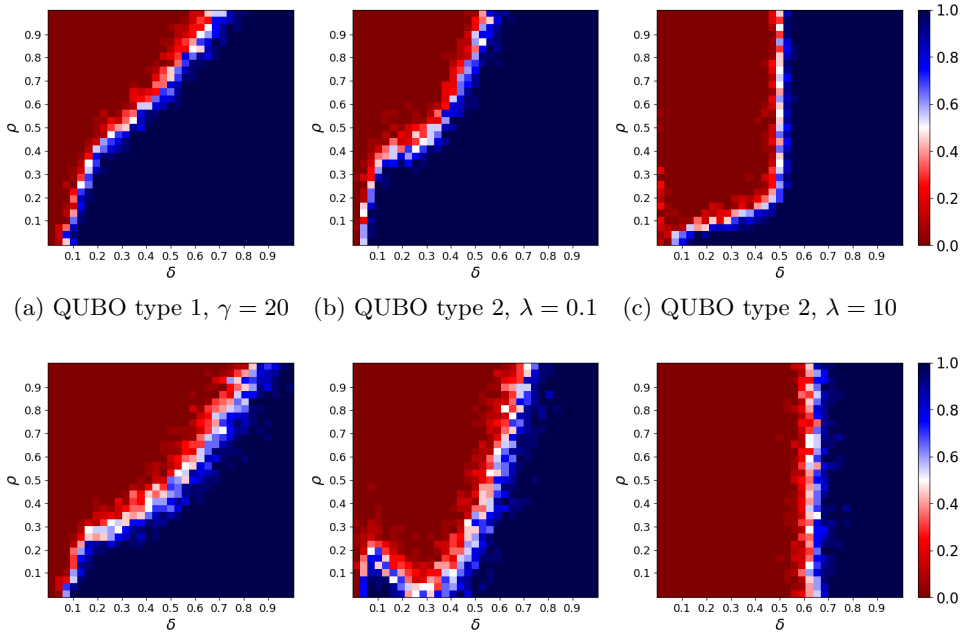
In this subsection we investigate whether the second QUBO formulation (QUBO type 2) presented in Section 3 exhibits a different and eventually improved performance in comparison to the QUBO type 1 employed so far. Recall that the two QUBO formulations presented are theoretically equivalent, when $\gamma = \frac{1}{\lambda}$. Figure 3b and Figure 3c show phase transition diagrams obtained with QBsolv and QUBO type 2 for two different λ values. The QUBO type 2 formulation with $\lambda = 0.1$ delivers a phase transition which is very similar to the QUBO type 1 formulation with $\gamma = 20$. The size of the successful recovey region is essentially identical, with slight trade-off in a few points. On the other hand, the QUBO type 2 formulation with $\lambda = 10$ results in a very different phase transition, which suggests that under this parameter setting, the QBsolv performance is mostly dependent on the number of measurements taken and not the sparsity rate. The same experiment was performed using simulated annealing. We see that equivalent formulations give the same results. QUBO type 2 with $\lambda = 0.1$ in Figure 3e yields similar phase transition as QUBO type 1 with $\gamma = 10$ in Figure 2c; the same holds for Figure 3f and Figure 2a). Based on experiments, the simulated annealing sampler does not seem to benefit from the second QUBO formulation.

4.3 Comparison of classical and quantum solvers

To compare the performance of different algorithms we decided to solve the binary compressive sensing problem with all the solvers described in Section 3. Since we had limited computation time on the D-Wave QPU, we chose smaller signal sizes, with $N = 20$ with $N = 40$, and performed only one trial at each step of the phase diagram. We also opted for Qubo formulation 2 with $\lambda = 10$ as parameter, since it seemed to yield a phase transition which was not dependent on the sparsity rate. The results of thse experiments can be visualized in Figure 4. For $N = 20$ we clearly see that Gurobi, together with the hybrid-quantum solvers, QBsolv and D-Wave hybrid, achieve the best performance having the largest region of successful signal recovery. The Matlab-$\mathcal{L}_1$ solver displays the expected behaviour, for both values of N which were considerd. Both simulated annealing and the quantum annealing on the D-Wave QPU, with some exception, can recover signals successfully if enough measurements, i.e., $M > 0.6N$ are performed. For $N = 40$, we notice a decrease in the performance of all methods. In particular, quantum annealing suffers from the lack of custom parameter tuning. In this case, it is Gurobi that achieves the largest success region, followed by QBsolv and DWave-hybrid which perform worse in the upper-left corner of the diagram, where there are very few measurements taken.

Finally, we note that all results presented so far may heavily be influenced by the lack of custom parameter optimization. It is likely that each QUBO-based solver has a different optimum QUBO formulation and penalty value that can depend on the specific measurement matrix. However, due to the lack of computation resources, we have not been able to perform such parameter tuning.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9



(a) QUBO type 1, $\gamma = 20$ (b) QUBO type 2, $\lambda = 0.1$ (c) QUBO type 2, $\lambda = 10$

(d) QUBO type 1, $\gamma = 20$ (e) QUBO type 2, $\lambda = 0.1$ (f) QUBO type 2, $\lambda = 10$

Fig. 3: Phase diagrams with QBsolv (a, b and c) and simulated annealing (d, e and f), $N = 300$, and 20 problems per step. The color indicate the probability that reconstruction for $(\delta = \frac{M}{N}, \rho = \frac{K}{M})$ instances is successful.

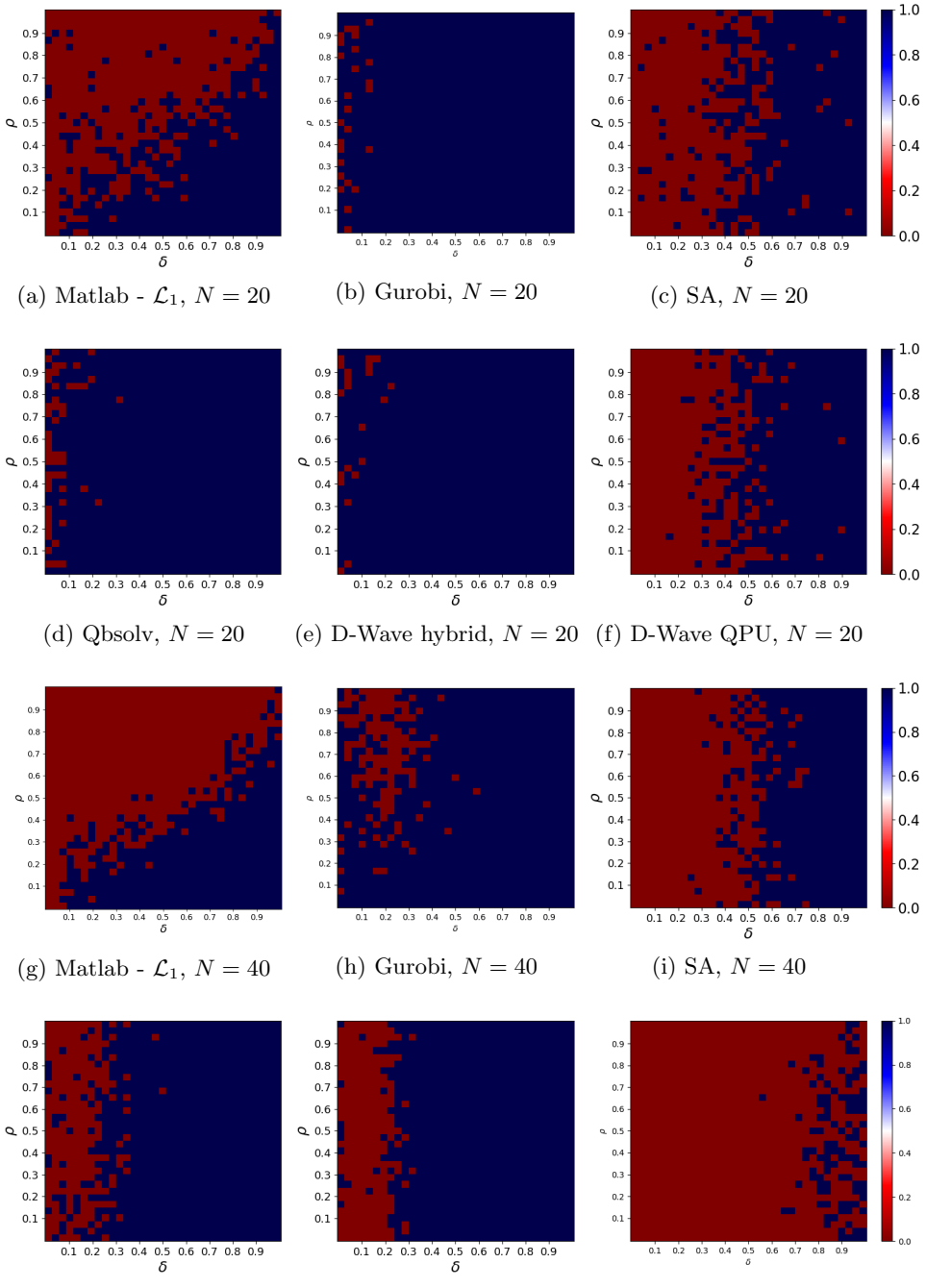
5 Conclusions

In this paper we tackled the binary compressive sensing problem and provide the first numerical results using a real quantum device. Different classical and (hybrid--)quantum QUBO solvers have been compared quantitatively to a classical $\mathcal{L}_1$-based approach by calculating phase-transition diagrams.

Based on the results presented in Section 4, we conclude that using a QUBO approach for solving the binary compressive sensing problem, the resulting phase-transition diagrams are significantly improved from the classical $\mathcal{L}_1$-based approach. For the Gaussian measurement matrices considered, a clear phase transition could be identified for all classical and (hybrid)-quantum solvers. In particular, for hybrid approaches such as QBsolv and D-Wave hybrid the phase transition was found to only depend on the undersampling rate, and not the sparsity of the signal. This is a positive result as the number of measurement taken is in principle a controllable quantity in signal recovery experiments.

For the small problem instances considered we see that the Gurobi solver slightly outperforms the hybrid-quantum annealing approaches. The hope is, once hardware grows and larger problems can be embedded on the quantum

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9



(a) Matlab - $\mathcal{L}_1$, $N = 20$ (b) Gurobi, $N = 20$ (c) SA, $N = 20$

(d) Qbsolv, $N = 20$ (e) D-Wave hybrid, $N = 20$ (f) D-Wave QPU, $N = 20$

(g) Matlab - $\mathcal{L}_1$, $N = 40$ (h) Gurobi, $N = 40$ (i) SA, $N = 40$

(j) Qbsolv, $N = 40$ (k) D-Wave hybrid, $N = 40$ (l) D-Wave QPU, $N = 40$

Fig. 4: Phase transition diagrams generated $N = 20$ and $N = 40$ using 1 problem instance per step. QUBO problems have been solved with QUBO type 2 with $\lambda = 10$. The color indicate the probability that reconstruction for ($\delta = \frac{M}{N}, \rho = \frac{K}{M}$) instances is successful.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

chip, that quantum annealing becomes a option when problem sizes become so large that they are unfeasible to solve using classical approaches such as Gurobi. Nevertheless our work shows the high potential of QUBO-based formulations compared to $\mathcal{L}_1$ minimization. The QUBOs presented in this paper can also be adapted to handle real or complex-valued through the appropriate usage of slack variables.

Currently, we do not have a full understanding yet on how to optimally select the different parameters (QUBO type, penalty value) for each solver. We expect that the choice of value for the penalty parameters γ or λ when using the QUBO formulation depends on ρ, δ and also the entries of the measurement matrix A. Hence, the only way to eventually infer an expression of γ or λ is to design a structured grid search, and evaluate a large set of values for each problem instance considered at each point in the phase diagram. This is one of the aspects we consider worthwhile for further investigation.

Acknowledgements This work was supported by the Dutch Ministry of Defense under Grant V2104.

References

1. D-Wave Ocean Software Documentation. https://docs.ocean.dwavesys.com/en/stable/, accessed: 2021-12-11
2. D-Wave Ocean Software Documentation: Simulated Annealing Sampler. https://docs.ocean.dwavesys.com/en/stable/docs_neal/reference/sampler.html#neal.sampler.SimulatedAnnealingSampler, accessed: 2021-12-05
3. D-Wave Problem Solving Handbook: Using Hybrid Solvers. https://docs.dwavesys.com/docs/latest/handbook_hybrid.html
4. D-Wave System online documentation: QPU-specific characteristics. https://docs.dwavesys.com/docs/latest/doc_physical_properties, accessed: 2021-12-09
5. D-Wave System online documentation: What is quantum annealing? https://docs.dwavesys.com/docs/latest/c_gs_2.html#getting-started-qa, accessed: 2022-01-14
6. State of Mathematical Optimization Report. Tech. rep., Gurobi Optimization (2021)
7. Anitori, L.: Compressive Sensing and Fast Simulations, applications to Radar Detection. Ph.D. thesis, TU Delft (2012)
8. Ayanzadeh, R., Halem, M., Finin, T.: An Ensemble Approach for Compressive Sensing with Quantum. arXiv e-prints arXiv:2006.04682 (Jun 2020)
9. Ayanzadeh, R., Mousavi, S., Halem, M., Finin, T.: Quantum Annealing Based Binary Compressive Sensing with Matrix Uncertainty. arXiv e-prints arXiv:1901.00088 (Dec 2018)
10. Baraniuk, R.G.: Compressive sensing [lecture notes]. IEEE Signal Processing Magazine **24**(4), 118–121 (2007)
11. Bontekoe, T.H., Neumann, N.M.P., Phillipson, F., Wezeman, R.S.: Quantum computing for radar and sonar information processing (2021), unpublished

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9

12. Booth, M., Reinhardt, S.P., Roy, A.: Partitioning Optimization Problems for Hybrid Classical/Quantum Execution. Tech. rep., D-Wave: The Quantum Computing Company (October 2018)
13. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning **3**(1), 1–122 (2011)
14. Candès, E., Romberg, J.: Sparsity and incoherence in compressive sampling. Inverse Problems **23**(3), 969–985 (apr 2007)
15. Candès, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure and Applied Mathematics **59**, 1207–1223 (2005)
16. Candès, E.J.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique **346**(9), 589–592 (2008)
17. Dekkers, A., Aarts, E.: Global optimization and simulated annealing. Mathematical Programming **50**(1-3), 367–393 (Mar 1991)
18. Donoho, D., Tanner, J.: Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **367**(1906), 4273–4293 (Nov 2009)
19. Donoho, D.: Compressed sensing. IEEE Transactions on Information Theory **52**(4), 1289–1306 (2006)
20. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum computation by adiabatic evolution. arXiv:quant-ph/0001106v1 (2000)
21. Hayashi, K., Nagahara, M., Tanaka, T.: A user's guide to compressed sensing for communications systems. IEICE Transactions on Communications **E96.B**(3), 685–712 (2013)
22. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse ising model. Physical Review E **58**, 5355–5363 (1998)
23. Maleki, A., Donoho, D.L.: Optimally tuned iterative reconstruction algorithms for compressed sensing. IEEE Journal of Selected Topics in Signal Processing **4**(2), 330–341 (2010)
24. McGeoch, C., Farré, P.: The Advantage System: Performance Update. Tech. rep., D-Wave: The Quantum Computing Company (October 2021)
25. McGeoch, C., Farré, P., Bernoudy, W.: D-Wave Hybrid Solver Service + Advantage: Technology Update . Tech. rep., D-Wave: The Quantum Computing Company (September 2020)
26. Rani, M., Dhok, S.B., Deshmukh, R.B.: A systematic review of compressive sensing: Concepts, implementations and applications. IEEE Access **6**, 4875–4894 (2018)
27. Romanov, E., Ordentlich, O.: On compressed sensing of binary signals for the unsourced random access channel. Entropy **23**(5), 605 (May 2021)
28. Shannon, C.: Communication in the presence of noise. Proceedings of the IRE **37**(1), 10–21 (jan 1949)
29. Shirvanimoghaddam, M., Li, Y., Vucetic, B., Yuan, J., Zhang, P.: Binary compressive sensing via analog fountain coding. IEEE Transactions on Signal Processing **63**(24), 6540–6552 (2015)
30. Tropp, J.A., Wright, S.J.: Computational methods for sparse solution of linear inverse problems. Proceedings of the IEEE **98**(6), 948–958 (2010)
31. Yang, J., Zhang, Y.: Alternating direction algorithms for $\mathcal{L}_1$-problems in compressive sensing. SIAM J. Sci. Comput. **33**, 250–278 (2011)
32. Zhang, Y.: User's Guide for YALL1: Your ALgorithms for L1 Optimization. Tech. rep., Rice University, Houston, Texas (2009)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_9