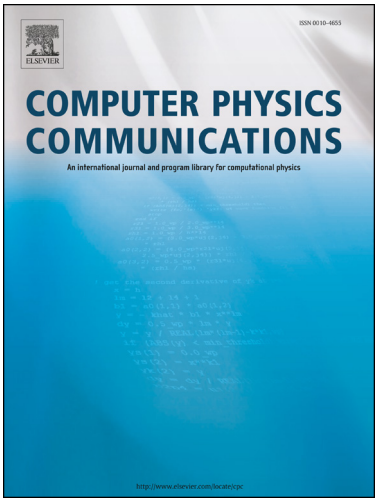

Computational overhead of locality reduction in binary optimization problems

Elisabetta Valiante, Maritza Hernandez, Amin Barzegar and Helmut G. Katzgraber

PII: S0010-4655(21)00214-9
DOI: https://doi.org/10.1016/j.cpc.2021.108102
Reference: COMPHY 108102

To appear in: *Computer Physics Communications*

Received date: 10 January 2021
Revised date: 6 July 2021
Accepted date: 15 July 2021

Please cite this article as: E. Valiante, M. Hernandez, A. Barzegar et al., Computational overhead of locality reduction in binary optimization problems, *Computer Physics Communications*, 108102, doi: https://doi.org/10.1016/j.cpc.2021.108102.

Computational overhead of locality reduction in binary optimization problems

Elisabetta Valiante[a,*], Maritza Hernandez[a], Amin Barzegar[b], Helmut G. Katzgraber[c,d,e,1]

[a] *1QB Information Technologies (1QBit), 1285 W Pender St Unit 200, Vancouver, British Columbia V6E 4B1, Canada*
[b] *Microsoft Quantum, Microsoft, Redmond, Washington 98052, USA*
[c] *Amazon Quantum Solutions Lab, Seattle, Washington 98170, USA*
[d] *AWS Intelligent and Advanced Compute Technologies, Professional Services, Seattle, Washington 98170, USA*
[e] *AWS Center for Quantum Computing, Pasadena, California 91125, USA*

Abstract

Recently, there has been considerable interest in solving optimization problems by mapping these onto a binary representation, sparked mostly by the use of quantum annealing machines. Such binary representation is reminiscent of a discrete physical two-state system, such as the Ising model. As such, physics-inspired techniques—commonly used in fundamental physics studies—are ideally suited to solve optimization problems in a binary format. While binary representations can be often found for paradigmatic optimization problems, these typically result in k-local higher-order unconstrained binary optimization cost functions. In this work, we discuss the effects of locality reduction needed for the majority of the currently available quantum and quantum-inspired solvers that can only accommodate 2-local (quadratic) cost functions. General locality reduction approaches require the introduction of ancillary variables which cause an overhead over the native problem. Using a parallel tempering Monte Carlo solver on Microsoft Azure Quantum, as well as k-local binary problems with planted solutions, we show that post reduction to a corresponding 2-local representation the problems become considerably harder to solve. We further quantify the

*Corresponding author
Email addresses: Elisabetta.Valiante@1qbit.com (Elisabetta Valiante), Maritza.Hernandez@1qbit.com (Maritza Hernandez), Amin.Barzegar@microsoft.com (Amin Barzegar), katzgrab@amazon.com (Helmut G. Katzgraber)

[1] The work of H. G. K. was performed before joining Amazon Web Services.

increase in computational hardness introduced by the reduction algorithm by measuring the variation of number of variables, statistics of the coefficient values, and the population annealing entropic family size. Our results demonstrate the importance of avoiding locality reduction when solving optimization problems.

Keywords:

PACS: 75.50.Lk, 75.40.Mg, 05.50.+q, 64.60.-i

1. Introduction

In recent years there have been many technological and algorithmic advances when solving optimization problems, in particular, in an industrial setting. Sparked by the work of D-Wave Systems Inc. [1, 2, 3], a whole new field of optimization based on physical processes has emerged. Some examples are: digital processors based on simulated annealing [4, 5, 6, 7] or other recently proposed algorithms [8, 9, 10, 11]; coherent Ising machines implemented with pulse lasers [12, 13, 14, 15, 16, 17], and other kinds of optical Ising machine [18, 19, 20, 21]; simulated bifurcation [22, 23] and other optimization using nonlinear oscillation networks [24, 25, 26, 27, 28]. Specifically, the development of hardware quantum annealers has stimulated new ways of tackling NP-hard problems previously inaccessible.

Despite these advances, the use of quantum annealers for large-scale industry applications remains limited if not paired with classical algorithms on CMOS hardware. Being able to tackle an application requires first having a Boolean representation of the problem. To this *mapping* step, in most cases a variable overhead is associated, which typically makes a problem harder to solve. However, due to hardware limitations, only 2-local (quadratic unconstrained binary optimization, or QUBO) cost functions can be tackled with quantum annealing hardware. This means that a higher-order binary polynomial unconstrained optimization problem requires a *locality reduction* which can result in a sizable variable overhead [29]. In this work, we focus on the *locality reduction*, and do not discuss additional overheads due to the *embedding* of a binary problem onto the hardwired sparse quasi-two-dimensional topology of annealing hardware or the effects of analog noise. See Refs. [29, 30] and Refs. [31, 32] for analyses

of overheads introduced by topology and analog noise, respectively.

The hardware limitations play an important role when solving problems naturally formulated as a Hamiltonian with k-local interactions with $k > 2$. There are various optimization problems in fundamental physics, computer science, and applications that are natively k-local. Examples in physics, as well as computer science, are computing the partition function of a four-dimensional pure lattice gauge theory [33, 34], measuring the fault-tolerance in topological colour codes [35], and solving k-SAT problems with $k > 2$. Examples of practical applications are circuit fault diagnosis [36, 29], molecular similarity measurement [37], molecular conformational sampling [38], and traffic light synchronization [39].

Quadratization techniques are algorithms used to reduce a higher-degree multilinear polynomial into a quadratic one [40]. The reduction process can introduce two different types of overheads. First, the quadratization itself can result in a large overhead before any solver is applied to the problem of interest. Nevertheless, the process is known to scale in polynomial time [41]. Second, quadratization requires the introduction of additional variables and terms. As such, the complexity of the problem increases and, in turn, so does the time to solution. Finally, the quadratization process might also introduce features (e.g., broader coupler distributions) that can affect the intrinsic difficulty of the problem. Both types of overheads might increase the complexity of the problem to the point of affecting its scaling. An extensive comparison between several quadratization methods, highlighting the pros and cons of each method, has been compiled by Dattani in Ref. [42].

In this paper, we use Microsoft Azure Quantum's k-local solvers based on simulated annealing and parallel tempering Monte Carlo to measure the time overhead introduced by the quadratization process to reduce an optimization problem with k-local interaction to its 2-local counterpart. We study unconstrained problems with a binary representation and planted solutions and disregard the time it takes for the quadratization algorithm to run. Our results demonstrate that the locality reduction introduces a large overhead when solving the problems. Employing a commonly used proxy metric, we demonstrate that, on average, optimization problems become much harder to solve when the locality is reduced. Reference [30] studies the embedding overhead when us-

ing sparse hardware topologies. Both complementary studies highlight the importance of developing new optimization machines and techniques that can handle k-local cost functions natively on complete graphs.

This paper has the following structure: in Sec. 2, we describe the benchmark problems used for the experiment; in Sec. 3, we present the setup of the experiment and the metrics used to compare performance; in Sec. 4 and Sec. 5, we discuss and analyze the results of the experiment; in Sec. 6, we present our conclusions.

2. Benchmark Problems

In order to study the computational overhead caused by reducing a k-local problem to a quadratic (2-local) formulation, we first generate Ising problems for $k = 3$ and $k = 4$. The k-local instances have been generated using the `Chook` package, which is publicly available on GitHub; see [43]. Using this package, we are able to construct planted-solution instances, thus ensuring that the ground state and corresponding energy are known a priori. The construction of k-local problems is performed by combining tile planting problems of lower-order, that is, $k \leq 2$.

The tile planting method [44, 45] decomposes the problem graph into edge-disjoint vertex-sharing subgraphs. It produces scalable problems with highly tunable complexity. `Chook` supports the generation of tile-planted problems on square and cubic lattice topologies with periodic boundary conditions. The regular structure of these lattices allows for a problem-graph decomposition that naturally renders a subset of the unit cells as subgraphs. Each subgraph is associated with an Ising cost Hamiltonian, and their sum defines the complete Hamiltonian of the problem.

The square lattices used in this work are defined by four subproblem classes that correspond to unit cycles (plaquettes) with different levels of frustration. A subproblem is constructed by assigning to the couplers values equal to -1, 1, or 2, according to the class to which the subproblem belongs. The class is assigned with a certain probability, and each instance class is defined by three probability parameters. We set the probability parameters to the default values used in `Chook`.

`Chook` constructs higher-order k-local problems ($k > 2$) by combining n Hamil-

tonians $\mathcal{H}^{(i)}$ with lower-order ($k \leq 2$) interactions and known ground states. If the Hamiltonians are completely independent of each other in that the underlying problem graphs do not share any vertices or edges, the composite Hamiltonian $\mathcal{H}_{\rm comp}$, obtained as the product of the $\mathcal{H}^{(i)}$, is minimized by any of the n known ground states. The locality of $\mathcal{H}_{\rm comp}$ is given by $k_{\rm max} = \sum_{i=1}^{n} k_{\rm max}^{(i)}$, with $k_{\rm max}^{(i)}$ being the locality of the highest-order term in the $\mathcal{H}^{(i)}$.

In this study, 3-local instances have been generated by combining a square tile planting problem with Ising spins coupled to a bimodal random field, while for 4-local instances, the problems have been generated by combining two square tile planting problems. For each locality considered, we generate instances with problem sizes N (number of variables) between 16 and 400.

The k-local instances are then reduced to their quadratic form using an iterative reduction-by-substitution algorithm [46, 41]. Here, we consider the terms in the problem with degree $k_t > 2$: we substitute the product of two binary variables with a new auxiliary variable and add a penalty term to enforce equality in the ground state.

A simple example of reducing a term is the following. Let us assume we have a third-degree binary polynomial with a term $x_1x_2x_3$, where we substitute $y = x_1x_2$ and introduce a penalty term:

$$\begin{aligned} & x_1x_2x_3 \implies \\ & \qquad yx_3 + C_{x_1,x_2}(x_1x_2 - 2x_1y - 2x_2y + 3y). \end{aligned} \tag{1}$$

The penalty term is always equal to 0 when the value of the auxiliary variable y is equal to the product of the binary variables x_1 and x_2. The constant C_{x_1,x_2} ensures that the constraint associated with the substitution of the product x_1x_2 is always satisfied. In fact, the constraint has to be obeyed regardless of the value of the other terms of the polynomial.

When reducing our problem instances, this process is repeated until the final function becomes quadratic. Tuning the value of the constants C_{x_a,x_b} is extremely important: a small value could return a 2-local problem not having the same optimum as the original higher-order problem. Therefore, a large value is commonly used in various implementations of this algorithm. As suggested in Refs. [46, 41], in a generic binary

polynomial $P(x_1, x_2, \ldots, x_n) = \sum_{i \in T} c_i t_i$, where t_i are the terms of the polynomial, a single constant can be defined for any substitution by summing all the coefficients:

$$C_{x_a, x_b} > \sum_{i \in T} |c_i|. \tag{2}$$

The absolute values of the coefficients, in a large polynomial, can accumulate to a very large number: this can pose issues when attempting to solve problems on current analog quantum annealing hardware, because large coefficients amplify the effects of the analog noise.

The reduction of k-local problems in this work is done via the `Hobo2Qubo` function available earlier through 1QBit's 1Qloud platform [47], which uses a tight bound for the penalty coefficient and sets it independently for each reduced term. The computational time required to reduce a single instance is negligible with respect to the time required by the solver. Moreover, the reduction from k-local to 2-local is known to scale in polynomial time [41], that is, it should be negligible for large problem sizes, relative to the exponential scaling of the cost of solving the problem.

The sizes and densities of the 2-local instances obtained after reduction from 3-local and 4-local instances are shown in Tables 1 and 2, respectively. The number of variables increases considerably when reducing locality from k-local to 2-local, as can be expected for a reduction-by-substitution algorithm.

The density of a k-local instance ρ is calculated as the sum of the densities for each degree in the polynomial, normalized for the number of degrees ≥ 2. The terms of the sum are calculated as the fraction of non-zero couplings over all the possible couplings for each degree. This is implemented as

$$\rho = \frac{1}{k-1} \sum_{k_\mathrm{t}=2}^{k} \frac{(N-k_\mathrm{t})! k_\mathrm{t}!}{N!} E_{k_\mathrm{t}}\,, \tag{3}$$

where k is the locality of the polynomial. The sum is taken over all the degrees in the polynomial running from $k_\mathrm{t} = 2$ to $k_\mathrm{t} = k$, E_{k_t} is the number of individual terms with degree k_t, and N is the number of variables in the polynomial. For 2-local instances, this expression is reduced to the common graph density expression. Tables 1 and 2 report the mean densities calculated over 30 instances for each problem size.

Notice that, for almost all problems, the densities decrease slightly after locality reduction. This is an interesting observation because, in general, a larger density is expected to be associated with larger complexity [7].

Table 1: Reduction of 3-local problems to 2-local problems. Densities for each instance are calculated as per Eq. (3). The mean values (denoted by an overbar) and their standard deviations are calculated over the 30 instances that have been generated. The number of variables of the reduced problems increases by a factor ~ 3.

3-local		2-local reduction	
N	$\bar{\rho}$	$\bar{N}$	$\bar{\rho}$
16	0.568 ± 0.020	46.73 ± 0.573	0.329 ± 0.009
64	0.398 ± 0.014	192.0	0.295 ± 0.003
144	0.364 ± 0.008	432.0	0.294 ± 0.002
256	0.352 ± 0.007	768.0	0.294 ± 0.002
400	0.345 ± 0.005	1200.0	0.294 ± 0.001

Table 2: Reduction of 4-local problems to 2-local problems. Densities for each instance are calculated as per Eq. 3. The mean values (denoted by an overbar) and their standard deviations are calculated over the 30 instances that have been generated. The number of variables of the reduced problems increases by a factor ~ 6.

4-local		2-local reduction	
N	$\bar{\rho}$	$\bar{N}$	$\bar{\rho}$
16	0.615 ± 0.023	76.5 ± 2.0	0.301 ± 0.013
64	0.295 ± 0.017	448.5 ± 4.0	0.167 ± 0.004
144	0.210 ± 0.008	887.6 ± 2.2	0.176 ± 0.002
256	0.179 ± 0.007	1501.3 ± 3.6	0.173 ± 0.002
400	0.163 ± 0.004	2248.3 ± 3.6	0.174 ± 0.001

3. Experiment Setup

The simulations are performed using Microsoft Azure Quantum's solvers, which can handle k-local terms natively. There are two variants of the solvers, parameter-free solvers and standard solvers. The parameter-free version requires the user to enter only a timeout and automatically optimizes the parameters to find solutions to binary cost functions to high probabilities. The standard solvers instead require parameter optimization to obtain the optimal performance.

3.1. Setup

For the experiments, we use the parameter-free `ParallelTempering` (v1.0) solver [48]. The best values for temperatures, number of sweeps, and number of replicas are calculated internally and are customized for each submitted problem individually. At the time the experiments discussed in this manuscripts are performed (July 2020), the solver does not disclose the parameters chosen for the optimization. The only parameter to set is `timeout`, which is the time spent in the core solver loop (in seconds). It is worth specifying that `timeout` does not include the time spent by the solver to calculate the parameters that are used during the optimization process. The total time the solver needs to solve the problem is referred to as `runtime`. The advantage of using a parameter-free solver is that no tuning experiment is necessary. The disadvantage is that the `runtime` we measure includes both the time to calculate the parameters and the time to solve the problem. At the time of running the experiment, the parameters calculated by the solver are not returned to the user in the current implementation. As such, we cannot list them in this work.

The benchmark experiment consists of solving 30 random instances for each system size and locality, as well as their respective 2-local reduction (see Tables 1 and 2 for details). For each of these instances, we perform 30 runs to gather statistics. We set `timeout` $= 100$. In cases when 100 is not enough time to find the ground-state energy, we increase `timeout` to 500.

3.2. Metrics

The primary objective of our benchmark experiment is to quantify how the computational effort in solving a problem scales as the size of the problem input increases. The common approach is to measure the time to solution (TTS). We calculate the TTS following the approach defined in Refs. [49, 7]:

$$\mathrm{TTS} = \tau \mathrm{R}_{99}, \tag{4}$$

where R_{99} is the number of runs required to find the ground-state energy with a probability of 99% and τ is the time it takes to run the algorithm once (i.e., the solver output `runtime`).

We derive R_{99} by estimating its distribution of the 50th percentile. This requires the algorithm to find the ground-state energy of each problem for at least 50% of the successive runs performed (see Ref. [7] for more details). When it is not possible to measure the TTS, because the ground-state energy cannot be determined sufficiently often, we measure other performance metrics, such as the fraction of solved problems and the residual energies—both defined below.

The fraction of solved problems is defined as the fraction of runs for which the ground-state energy is found by the solver divided by the total number of experiments. We have performed a total of 900 runs for each problem size and locality. The energy is calculated for each problem and each run in the following way:

$$R = \frac{E_{\mathrm{GS}} - E_{\mathrm{best}}}{E_{\mathrm{GS}}}, \tag{5}$$

where E_{GS} is the known planted ground-state energy of the problem and E_{best} is the best energy found by the algorithm. The values reported here are obtained by resampling the distribution of residuals over all problems and runs.

4. Results

Figure 1 shows the TTS for planted 3- and 4-local problems with a number of variables N ranging from 16 to 400 using the parallel tempering algorithm. Both problem types show a similar scaling. We have fit an exponential function of the form

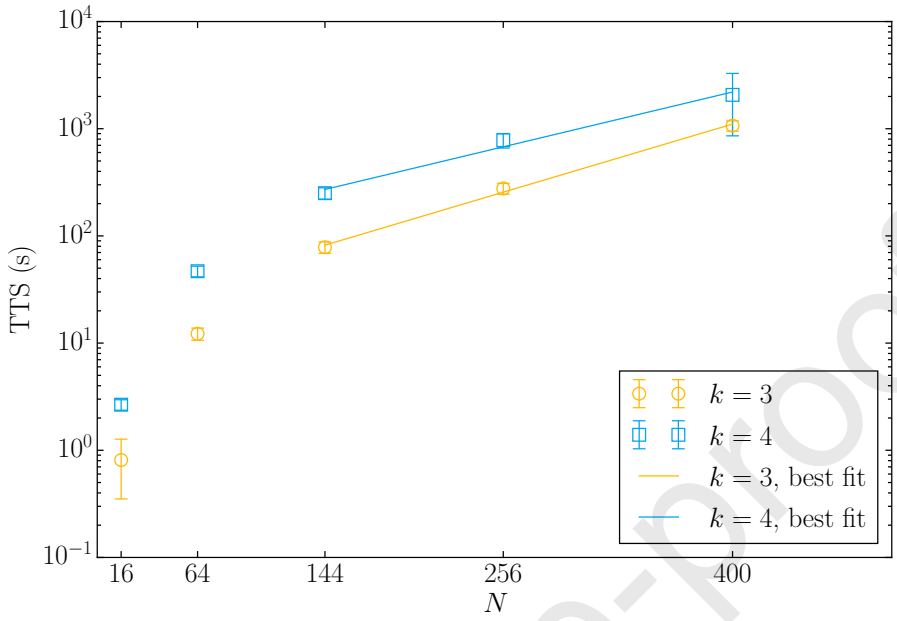

Figure 1: TTS mean value for k-local problems with $k = 3$ and $k = 4$ using the parallel tempering solver. The error bars correspond to a 2σ confidence interval. The continuous lines show the result of fitting an exponential function over the three largest problem sizes (see text for more details).

$\mathrm{TTS} = 10^{\alpha+\beta N}$ over the three largest problem sizes. The results of the fit and the estimated scaling exponent β are:

$$\beta = 0.00441(14) \qquad (k = 3)$$
$$\beta = 0.00355(67) \qquad (k = 4)$$

The fraction of solved runs is 100% for all sizes of both 3- and 4-local problems. However, it is not possible to calculate the TTS for the 2-local reductions of either the 3- or 4-local problems. Figure 2 shows the fraction of solved problems (top panel) and the residual energies (bottom panel). The 2-local problems derived from the 4-local instances seem to have a larger overhead than the ones generated from the 3-local problems, but it is not clear if they have a different computational cost scaling with system size. We surmise that the higher the locality, the larger such overhead in solving the 2-local reductions will be. The benchmark experiment has been performed

with two different values of the parameter `timeout`. However, increasing the timeout does not improve the quality of the results.

5. Discussion

The computational hardness of the 3- and 4-local instances is set in the planting tool `Chook` by a careful choice of the couplers from different disorder distributions with varying levels of frustration. The reduction to 2-local interactions in the Hamiltonian requires the introduction of auxiliary variables and penalty terms. Tables 1 and 2 show the increase in the number of variables when reducing the problems to their 2-local versions. We observe an increase of a factor of approximately 3 for the 3-local problems, which increases to a factor ~ 6 when reducing the 4-local problems. Higher-order Hamiltonians will naturally require an even larger overhead.

Figure 3 compares the coupler distributions for the 3- and 4-local problems of different system sizes with their corresponding 2-local reductions. The histograms show that, while the distributions of the 3- and 4-local problem are quite similar (note that the x-axis in the plots on the left and the right sides of the figure use a different scale), the distributions of their 2-local reductions are significantly wider, in particular when the reduction occurs from a higher degree of the polynomial, and no longer symmetric. A more quantitative analysis of this effect is shown in Figure 4. We have calculated the standard deviation and the kurtosis of the the coupler distributions. While the former increases by a factor of approximately 10, the latter reduces by approximately a factor of 5 when reducing the problems from k-local to 2-local. Having a large dynamic range in the coupler distributions of the reduced problems typically makes these harder to solve with physics-based solvers.

The introduction of penalty terms shifts the mean of the coupler distribution by increasing the weight of large positive values: this is an indicator that the frustration will be affected. To confirm this intuition, we have measured the level of frustration for the 3- and 4-local instances and their 2-local reductions. A misfit parameter is used to characterize the degree of frustration of ordered and disordered systems. It measures the increase of the ground-state energy due to frustration, in comparison with that of a

relevant reference state. We have measured the level of frustration by calculating the misfit parameter as suggested in [50]:

$$\mu_0 = \frac{E_0 - E^{\mathrm{id}}_{\mathrm{min}}}{E^{\mathrm{id}}_{\mathrm{max}} - E^{\mathrm{id}}_{\mathrm{min}}}, \tag{6}$$

where E_0 is the ground energy of our instances, and $E^{\mathrm{id}}_{\mathrm{min}}$ and $E^{\mathrm{id}}_{\mathrm{max}}$ describe the minimal and maximal ideally possible energy values, respectively, where "ideal" refers to the assumption that all local energies yields a minimal (and maximal) contribution to the total energy. These energies are calculated assuming that all bonds are satisfied (and non-satisfied). Figure 5 presents the average values of the misfit parameter as a function of the problem size calculated over all instances generated for each locality. We observe that the reduction process generates an increase in the degree of frustration level of about 3% for 3-local instances, and an increase of about 1% for 4-local instances. The increase in frustration does not show any dependence on the problem size.

Our conclusion is that the locality reduction makes the problem computationally harder, possibly as a combination of the increase of the number of variables, the greater variance in the coupler distribution, and the change in the frustration level.

To corroborate the aforementioned observation we use population annealing Monte Carlo (PAMC) [51, 52, 53, 54, 55, 56] to measure the *entropic family size* ρ_{s}. Similar to simulated annealing (SA) [57], population annealing is a sequential Markov chain Monte Carlo (MCMC) algorithm in which a population of "replicas" is slowly annealed toward a target low temperature. At each temperature, the population is reconfigured via a resampling process during which some replicas are multiplied or eliminated to achieve an equilibrium Gibbs distribution of energies. In a well-thermalized PAMC simulation, a sufficient number of the original replica families must survive. This can be quantified by the *family entropy*, S_{f},

$$S_{\mathrm{f}} = -\sum_{i}^{R} \mathfrak{n}_i \log \mathfrak{n}_i \,, \tag{7}$$

where $\mathfrak{n}_i$ is the fraction of the replicas in the i-th family and R is the total population size. Large fluctuations in the resampling are a signature of the difficulty in attaining thermal equilibrium, and lead to the descendants of only a few original copies dominating the population [52]. Hence, a measure of the effective number of surviving

replica families at the lowest temperature allows one to distinguish between hard and easy problems. The *entropic family size* in thermal equilibrium is defined as

$$\rho_{\rm s} = \lim_{R\to\infty} R/e^{S_{\rm f}}. \tag{8}$$

For a given set of simulation parameters, the larger the value of $\rho_{\rm s}$, the smaller the number of surviving families, and the more rugged the problem's energy landscape will be. Thus, $\rho_{\rm s}$ provides a measure of hardness for algorithms that are based on local search in the classical energy landscape. As shown in Ref. [45], $\rho_{\rm s}$ is highly correlated with other well-established hardness metrics, such as the integrated autocorrelation time in parallel tempering Monte Carlo [54]. Note that $\rho_{\rm s}$, by definition, is an intensive quantity, and therefore independent of the population size R in the thermodynamic limit. In practice, $\rho_{\rm s}$ converges to its true value at a large but finite population.

For all PAMC simulations, we use a linear schedule in inverse temperature. The linear schedule runs between $1/T_{\rm start} = 0$ and $1/T_{\rm end} = 20$ in 100 steps for smaller problems and up to 300 steps for larger ones. In each problem, the coupler values are normalized by the maximum energy scale, such that the same $T_{\rm end}$ can be used for all the benchmarking problems. We perform 10 Metropolis sweeps per replica at each temperature. For all measurements of $\rho_{\rm s}$, we ensure that convergence is achieved unless the simulation times out. We use the following procedure to determine the convergence of the PAMC simulations on each problem instance: starting from a relatively small population size (such as $R = 8$), we run PAMC 100 times and record the value of $\rho_{\rm s}$ at the final temperature $T_{\rm end}$ for each restart. The mean value and the corresponding error for each problem are then calculated using the set of computed $\rho_{\rm s}$ values. Then, we double R and repeat the above process. A satisfactory convergence is reached when consecutive values of $\rho_{\rm s}$ agree within the errors.

Figure 6 shows the mean and standard deviation of $\rho_{\rm s}$ over all instances generated for each problem size, for different values of R. The value of R at which $\rho_{\rm s}$ converges depends on the size and hardness of the problem instances and can be visualized as a plateau in the curves. We can observe how $\rho_{\rm s}$ converges to values of ~ 10 at a population of $R \simeq 50$ for 3- and 4-local problems, while it converges to much higher values for the 2-local problems. On average, $\rho_{\rm s}$ converges at $R \simeq 10^3 - 10^4$ for 3-local

reduced instances and $R > 10^4$ for 4-local reduced ones. In particular, measuring ρ_s for the reduced version of 4-local problems is possible for sizes $N = 16$ and $N = 64$ only. The problems are so hard that the simulation converges only partially for $N = 144$ (meaning that not all problem instances converge), and does not converge at all during the allocated time for larger problem sizes. Figure 7 shows the converging values of ρ_s we obtain from the simulation for each system size. The 2-local reduction critically increases the hardness of the problems, especially for large system sizes.

Our results demonstrate the advantage of solving the optimization problems in their original k-local formulation, and we expect this result to be independent of the choice of solver. Reference [29] shows that a simulated quantum annealing (SQA) algorithm has no advantage in solving a k-local formulation of a problem, instead of its 2-local reduction, but the study includes only instances with $N < 20$.

6. Conclusions

We have generated problems with planted solutions having k-local interactions and reduced them to their corresponding 2-local versions, more amenable to current physics-inspired optimization tools than the original ones. The reduction has been performed using a customized version of a classic and extensively adopted quadratization algorithm. The computational time required by the reduction algorithm is known to scale polynomially with the size of the input and thus does not affect the overall exponential scaling found in current physics-inspired optimization methods. Using Microsoft Azure Quantum's implementation of the `ParallelTempering` parameter-free algorithm, designed to handle problems of any locality, we have attempted to find optima for the native 3- and 4-local problems, as well as their 2-local reductions. All k-local problems with $k = 3$ and $k = 4$ have been solved to optimality during the allocated 100-second timeout. The TTS for 4-local problems is approximately 5 times larger than for the 3-local ones. In contrast, even after increasing the timeout to 500 seconds, the 2-local reductions could not be solved. It is common practice to apply locality reduction in order to accommodate higher-order polynomial unconstrained optimization problems to run on optimizers that natively handle only quadratic problems,

such as the D-Wave quantum annealer, the Fujitsu Digital Annealer, or the Toshiba Simulated Bifurcation Machine. Nevertheless, our results show that doing so should ideally be avoided. As such, investing into creating hardware and/or software to tackle higher-order problems should be prioritized.

7. Acknowledgments

We thank the two anonymous referees for their insightful comments and valuable suggestions that allowed us to considerably improve our analysis. We thank Marko Bucyk for his careful editing and reviewing of the manuscript. H. G. K. would like to thank David Poulin for inspiring discussions and dedicate this manuscript to him.

References

[1] M. Johnson, M. Amin, S. Gildert, et al., Quantum annealing with manufactured spins, Nature 473 (2011) 194–198.

[2] N. Dickson, M. Johnson, M. Amin, et al., Thermally assisted quantum annealing of a 16-qubit problem, Nat. Commun. 4 (1903) (2013) 1903.

[3] P. I. Bunyk, E. M. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, A. J. Przybysz, J. Whittaker, Architectural considerations in the design of a superconducting quantum annealing processor, IEEE Trans. Appl. Supercond. 24 (4) (2014) 1–10.

[4] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, H. Mizuno, A 20k-Spin Ising Chip to Solve Combinatorial Optimization Problems With CMOS Annealing, IEEE Journal of Solid-State Circuits 51 (1) (2016) 303–309.

[5] K. Yamamoto, W. Huang, S. Takamaeda-Yamazaki, M. Ikebe, T. Asai, M. Motomura, A Time-Division Multiplexing Ising Machine on FPGAs, Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies (3) (2017) 1–6.

[6] S. Tsukamoto, M. Takatsu, S. Matsubara, H. Tamura, An Accelerator Architecture for Combinatorial Optimization Problems, FUJITSU Sci. Tech. J. 53 (2017) 8–13.

[7] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, H. G. Katzgraber, Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer, Frontiers in Physics 7 (2019) 48.

[8] T. Okuyama, T. Sonobe, K. Kawarabayashi, M. Yamaoka, Binary optimization by momentum annealing, Phys. Rev. E 100 (2019) 012111.

[9] S. Patel, L. Chen, P. Canoza, S. Salahuddin, Ising Model Optimization Problems on a FPGA Accelerated Restricted Boltzmann Machine (2020). arXiv:2008.04436.

[10] K. Yamamoto, K. Kawamura, K. Ando, N. Mertig, T. Takemoto, M. Yamaoka, H. Teramoto, A. Sakai, S. Takamaeda-Yamazaki, M. Motomura, STATICA: A 512-Spin 0.25M-Weight Annealing Processor With an All-Spin-Updates-at-Once Architecture for Combinatorial Optimization With Complete Spin–Spin Interactions, IEEE Journal of Solid-State Circuits 56 (1) (2021) 165–178.

[11] T. Leleu, F. Khoyratee, T. Levi, R. Hamerly, T. Kohno, K. Aihara, Scaling advantage of nonrelaxational dynamics for high-performance combinatorial optimization (2021). arXiv:2009.04084.

[12] Z. Wang, A. Marandi, K. Wen, R. L. Byer, Y. Yamamoto, Coherent Ising machine based on degenerate optical parametric oscillators, Phys. Rev. A 88 (2013) 063853.

[13] A. Marandi, Z. Wang, K. Takata, R. L. Byer, Y. Yamamoto, Network of time-multiplexed optical parametric oscillators as a coherent Ising machine, Nat. Photonics 8 (12) (2014) 937–942.

[14] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara,

K.-i. Kawarabayashi, K. Inoue, S. Utsunomiya, H. Takesue, A coherent Ising machine for 2000-node optimization problems, Science 354 (6312) (2016) 603–606.

[15] P. L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsunomiya, K. Aihara, R. L. Byer, M. M. Fejer, H. Mabuchi, Y. Yamamoto, A fully programmable 100-spin coherent Ising machine with all-to-all connections, Science 354 (6312) (2016) 614–617.

[16] Y. Yamamoto, K. Aihara, T. Leleu, K.-i. Kawarabayashi, S. Kako, M. Fejer, K. Inoue, H. Takesue, Coherent Ising machines—optical neural networks operating at the quantum limit, npj Quantum Inf. 3 (1) (2017) 49.

[17] R. Hamerly, T. Inagaki, P. L. McMahon, D. Venturelli, A. Marandi, T. Onodera, E. Ng, C. Langrock, K. Inaba, T. Honjo, K. Enbutsu, T. Umeki, R. Kasahara, S. Utsunomiya, S. Kako, K.-i. Kawarabayashi, R. L. Byer, M. M. Fejer, H. Mabuchi, D. Englund, E. Rieffel, H. Takesue, Y. Yamamoto, Experimental investigation of performance differences between coherent Ising machines and a quantum annealer, Science Advances 5 (5) (2019).

[18] D. Pierangeli, G. Marcucci, C. Conti, Large-Scale Photonic Ising Machine by Spatial Light Modulation, Phys. Rev. Lett. 122 (21) (2019) 213902.

[19] D. Pierangeli, G. Marcucci, D. Brunner, C. Conti, Noise-enhanced spatial-photonic Ising machine, Nanophotonics 9 (13) (2020) 4109–4116.

[20] D. Pierangeli, G. Marcucci, C. Conti, Adiabatic evolution on a spatial-photonic Ising machine, Optica 7 (11) (2020) 1535–1543.

[21] D. Pierangeli, M. Rafayelyan, C. Conti, S. Gigan, Scalable Spin-Glass Optical Simulator, Phys. Rev. Applied 15 (2021) 034087.

[22] H. Goto, K. Tatsumura, A. R. Dixon, Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems, Science Advances 5 (4) (2019) eaav2372.

[23] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, K. Tatsumura, High-performance combinatorial optimization based on classical mechanics, Science Advances 7 (6) (2021) eabe7953.

[24] H. Goto, Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network, Sci. Rep. 6 (1) (2016) 21686.

[25] S. E. Nigg, N. Lörch, R. P. Tiwari, Robust quantum optimizer with full connectivity, Science Advances 3 (4) (2017) e1602273.

[26] S. Puri, C. K. Andersen, A. L. Grimsmo, A. Blais, Quantum annealing with all-to-all connected nonlinear oscillators, Nat. Commun. 8 (1) (2017) 15785.

[27] H. Goto, Z. Lin, Y. Nakamura, Boltzmann sampling from the Ising model using quantum heating of coupled nonlinear oscillators, Sci. Rep. 8 (1) (2018) 7154.

[28] H. Goto, Quantum Computation Based on Quantum Adiabatic Bifurcations of Kerr-Nonlinear Parametric Oscillators, J. Phys. Soc. Jpn. 88 (6) (2019) 061015.

[29] A. Perdomo-Ortiz, A. Feldman, A. Ozaeta, S. V. Isakov, Z. Zhu, B. O'Gorman, H. G. Katzgraber, A. Diedrich, H. Neven, J. de Kleer, et al., Readiness of Quantum Optimization Machines for Industrial Applications, Phys. Rev. Applied 12 (1) (2019) 014004.

[30] M. S. Könz, W. Lechner, H. G. Katzgraber, M. Troyer, Scaling overhead of embedding optimization problems in quantum annealing (2021). arXiv:2103.15991.

[31] Z. Zhu, A. J. Ochoa, S. Schnabel, F. Hamze, H. G. Katzgraber, Best-case performance of quantum annealers on native spin-glass benchmarks: How chaos can affect success probabilities, Phys. Rev. A 93 (2016) 012317.

[32] T. Albash, V. Martin-Mayor, I. Hen, Analog errors in Ising machines, Quantum Sci. Technol. 4 (2) (2019) 02LT03.

[33] G. D. las Cuevas, W. Dür, M. V. den Nest, H. J. Briegel, Completeness of classical spin models and universal quantum computation, J. Stat. Mech. 2009 (07) (2009) P07001.

[34] G. D. las Cuevas, W. Dür, H. J. Briegel, M. A. Martin-Delgado, Mapping all classical spin models to a lattice gauge theory, New J. Phys. 12 (4) (2010) 043014.

[35] R. S. Andrist, H. G. Katzgraber, H. Bombin, M. A. Martin-Delgado, Tricolored lattice gauge theory with randomness: fault tolerance in topological color codes, New J. Phys. 13 (8) (2011) 083006.

[36] A. Feldman, G. Provan, A. Van Gemund, Approximate Model-Based Diagnosis Using Greedy Stochastic Search, Journal of Artificial Intelligence Research 38 (2010) 371–413.

[37] M. Hernandez, A. Zaribafiyan, M. Aramon, M. Naghibi, A Novel Graph-Based Approach for Determining Molecular Similarity (2016). arXiv:1601.06693.

[38] D. J. J. Marchand, M. Noori, A. Roberts, G. Rosenberg, B. Woods, U. Yildiz, M. Coons, D. Devore, P. Margl, A Variable Neighbourhood Descent Heuristic for Conformational Search Using a Quantum Annealer, Sci. Rep. 9 (2019) 13708.

[39] Microsoft Quantum, Jij and Toyota Tsusho: reducing carbon emissions with Azure Quantum, https://cloudblogs.microsoft.com/quantum/2020/08/04/jij-toyota-azure-quantum-reducing-carbon-emissions/.

[40] E. Boros, A. Gruber, On Quadratization of Pseudo-Boolean Functions (2014). arXiv:1404.6538.

[41] E. Boros, P. L. Hammer, Pseudo-Boolean optimization, Discrete Applied Mathematics 123 (1) (2002) 155.

[42] N. Dattani, Quadratization in discrete optimization and quantum mechanics (2019). arXiv:1901.04405.

[43] D. Perera, I. Akpabio, F. Hamze, S. Mandrà, N. Rose, M. Aramon, H. G. Katzgraber, Chook – A comprehensive suite for generating binary optimization problems with planted solutions (2020). arXiv:2005.14344.

[44] F. Hamze, D. C. Jacob, A. J. Ochoa, D. Perera, W. Wang, H. G. Katzgraber, From near to eternity: Spin-glass planting, tiling puzzles, and constraint-satisfaction problems, Phys. Rev. E 97 (4) (Apr 2018).

[45] D. Perera, F. Hamze, J. Raymond, M. Weigel, H. G. Katzgraber, Computational hardness of spin-glass problems with tile-planted solutions, Phys. Rev. E 101 (2) (Feb 2020).

[46] I. Rosenberg, Reduction of bivalent maximization to the quadratic case, Cahiers du Centre d'Etudes de Recherche Operationnelle 17 (1975) 71.

[47] 1QBit, 1Qloud Documentation: Convert HOBO to QUBO, https://portal.1qbit-prod.com/docs/task/convert-hobo-to-a-qubo, accessed: 2020-12-09.

[48] Microsoft Quantum, Parallel Tempering, https://docs.microsoft.com/en-ca/azure/quantum/optimization-parallel-tempering, accessed: 2021-05-14.

[49] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, M. Troyer, Defining and detecting quantum speedup, Science 345 (6195) (2014) 420–424.

[50] S. Kobe, T. Klotz, Frustration: How it can be measured, Phys. Rev. E 52 (5) (1995) 5660–5663.

[51] K. Hukushima, Y. Iba, Population Annealing and Its Application to a Spin Glass, in: J. E. Gubernatis (Ed.), The Monte Carlo method in the physical sciences: celebrating the 50th anniversary of the Metropolis algorithm, Vol. 690, AIP, Los Alamos, New Mexico (USA), 2003, p. 200.

[52] J. Machta, Population annealing with weighted averages: A Monte Carlo method for rough free-energy landscapes, Phys. Rev. E 82 (2010) 026704.

[53] J. Machta, R. Ellis, Monte Carlo Methods for Rough Free Energy Landscapes: Population Annealing and Parallel Tempering, J. Stat. Phys. 144 (2011) 541.

[54] W. Wang, J. Machta, H. G. Katzgraber, Population annealing: Theory and application in spin glasses, Phys. Rev. E 92 (2015) 063307.

[55] C. Amey, J. Machta, Analysis and optimization of population annealing, Phys. Rev. E 97 (2018) 033301.

[56] A. Barzegar, C. Pattison, W. Wang, H. G. Katzgraber, Optimization of population annealing Monte Carlo for large-scale spin-glass simulations, Phys. Rev. E 98 (2018) 053308.

[57] S. Kirkpatrick, C. D. Gelatt, Jr., M. Vecchi, Optimization by Simulated Annealing, Science 220 (1983) 671.

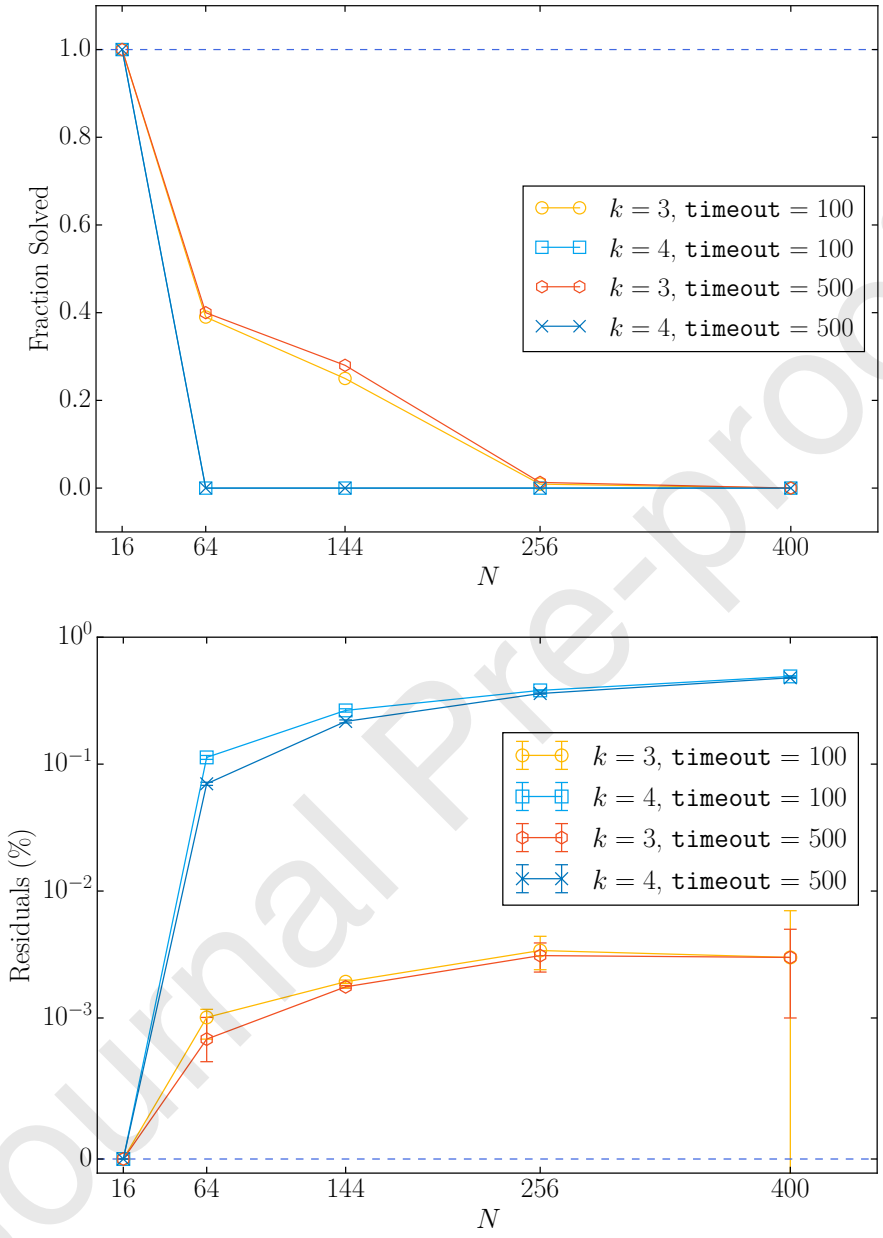

Figure 2: Fraction solved (top) and residuals (bottom) of 2-local problems obtained by reducing k-local instances with $k = 3$ and $k = 4$. The dashed line represents the reference for the ideal cases. The benchmark experiment has been performed with two different values of the parameter `timeout`. The data show that solving the 2-local versions of the problems is extremely difficult. In fact, we were unable to do a scaling analysis as the majority of the problems could not be solved.

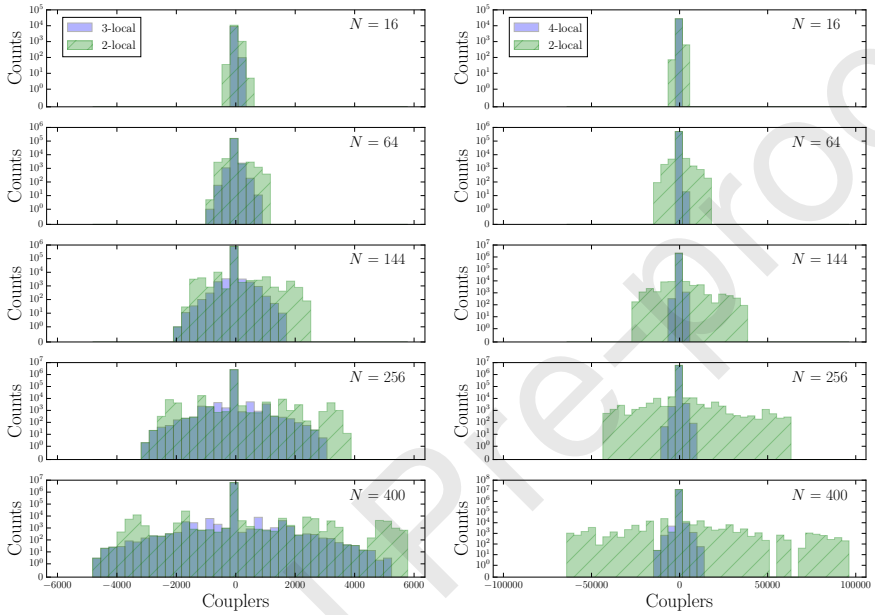

Figure 3: Coupler distributions of k-local problems with $k = 3$ (left panel) and $k = 4$ (right panel) for different system sizes N, and their corresponding 2-local reductions. The distributions of the 3- and 4-local problems are quite similar (note that the x-axis in the plots on the left and the right sides of the figure use a different scale). When comparing the original problems with their reduced form, we observe that,

in the 3-local case, the distributions are comparable, however, weight is redistributed to the tails. In the 4-local case, there is a sizable increase in the width of the distributions after locality reduction.

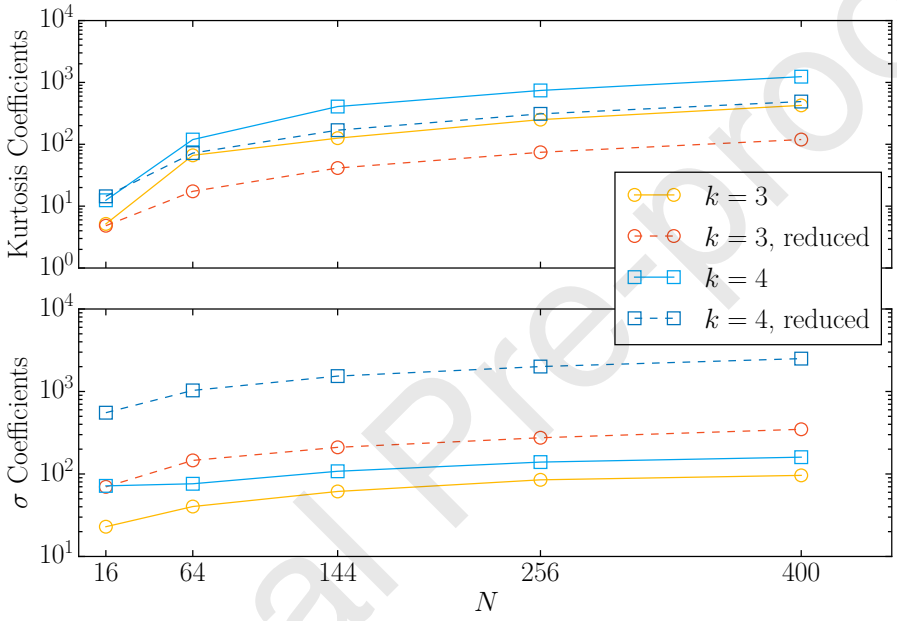

Figure 4: Kurtosis and standard deviation calculated from the coupler distributions of k-local problems with $k = 3$ and $k = 4$, and their correspondent 2-local reductions. While the kurtosis decreases, the standard deviation of the distributions increases noticeably, thus making the problems harder to solve. Both panels have the same horizontal axis.

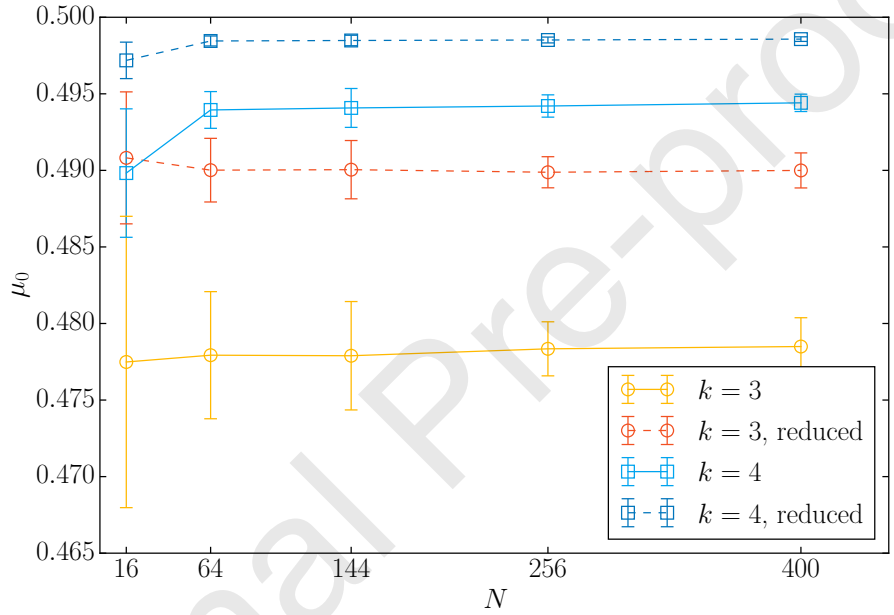

Figure 5: Average misfit parameter μ_0, calculated from the k-local instances with $k = 3$ and $k = 4$, and their correspondent 2-local reductions. The locality reduction increases the degree of frustration up to 3%.

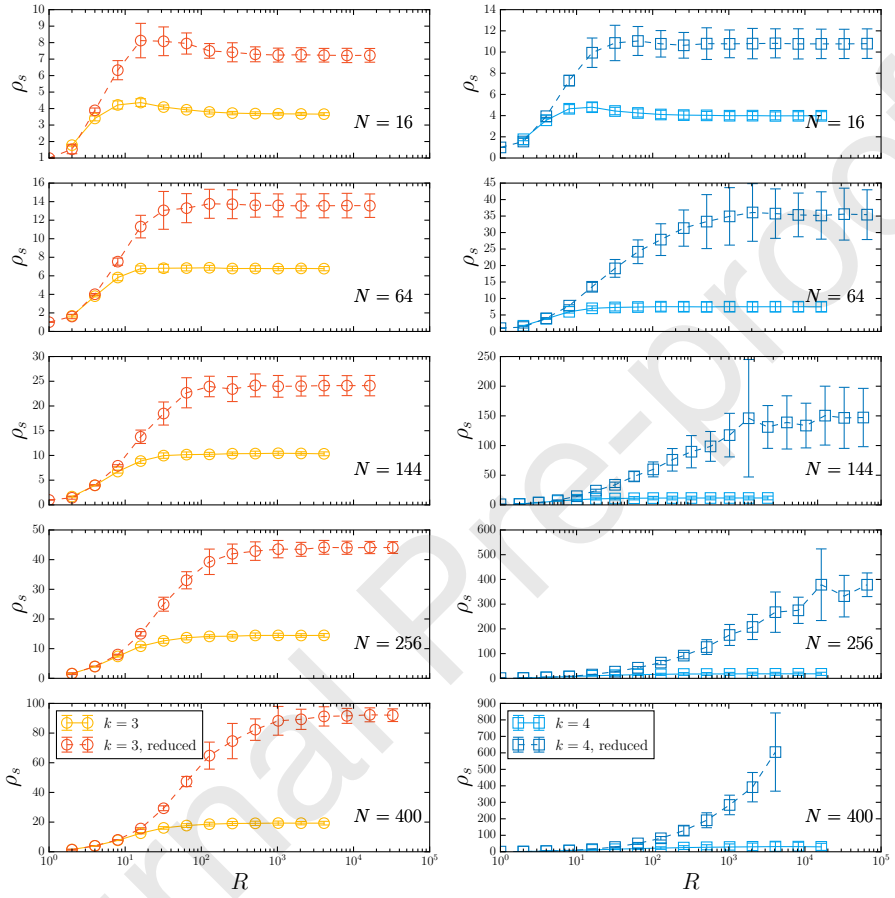

Figure 6: Entropic family size, ρ_s, calculated using population annealing Monte Carlo for k-local problems with $k = 3$ (left panel) and $k = 4$ (right panel), for different system sizes N, and their corresponding 2-local reductions. The family size is calculated by averaging over all instances generated for each system size for different values of R. The family size of the reduced version of $k = 4$ problems with $N = 144$ converges only partially, while for larger problem sizes the value does not converge during the allocated timeout.

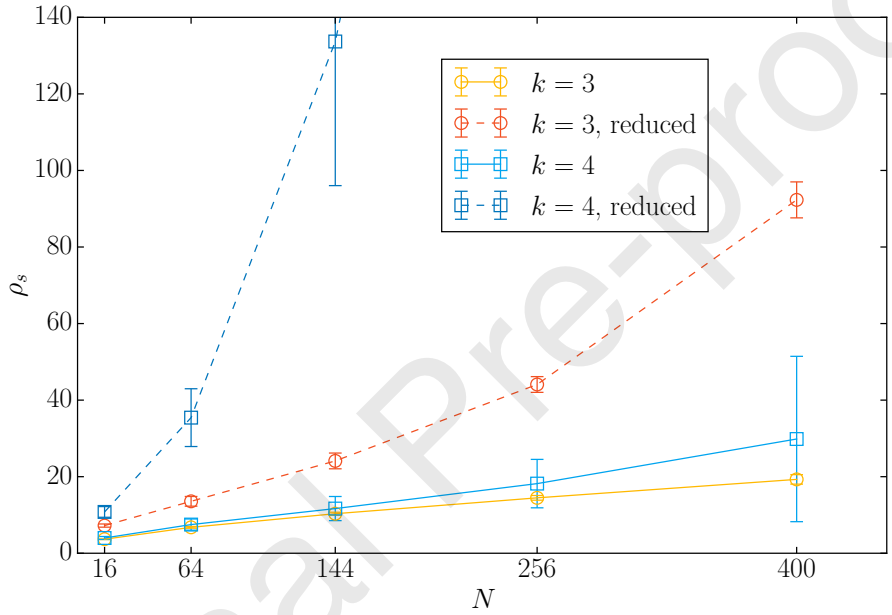

Figure 7: Entropic family size, ρ_s, calculated using population annealing Monte Carlo for k-local problems with $k = 3$ and $k = 4$, and their corresponding 2-local reductions. The family size confirms that a reduction in locality makes the problems computationally harder to solve.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: