# Spinto: High-performance Energy Minimization in Spin Glasses

Héctor J. García    Igor L. Markov
hjgarcia@eecs.umich.edu    imarkov@eecs.umich.edu
The University of Michigan,
2260 Hayward Street,
Ann Arbor, Michigan, 48109-2121, USA

***Abstract*—With the prospect of atomic-scale computing, we study cumulative energy profiles of spin-spin interactions in non-ferromagnetic lattices (Ising spin-glasses)—an established topic in solid-state physics that is now becoming relevant to atomic-scale EDA. Recent proposals suggest non-traditional computing devices based on nature's ability to find min-energy states. *Spinto* utilizes EDA-inspired high-performance algorithms to (i) simulate natural energy minimization in spin systems and (ii) study its potential for solving hard computational problems. Unlike previous work, our algorithms are not limited to planar Ising topologies. In one CPU-day, our branch-and-bound algorithm finds min-energy (ground) states on $100$ spins, while our local search approximates ground states on $1,000,000$ spins. We use this computational tool to study the significance of *hyper-couplings* in the context of recently implemented adiabatic quantum computers.**

## I. Introduction

As leading CMOS foundaries are gearing for mass production of 22nm and 32nm CMOS chips, long-term EDA research has started to explore the use of atomic properties in computing. This exploration relies on established computational models of atomic-scale phenomena, but struggles to connect different levels of abstraction—spin-level micro models and energy-based statistical macro models.

The spin-glass model was proposed by Edwards and Anderson [6] as a variation of the Ising model to study disorder in crystallized solids. Such systems are composed of particles that can be in either of two possible energy *spin* states. The model is described in graph-theoretic terms by representing atoms in a crystal with vertices and bonds between atoms with edges. Since physical systems found in nature are often disordered, the edges are assigned random weight values corresponding to the energy of the bond. Some physical and chemical properties of a crystal depend on the total energy of the bonds, which depend on atomic states. Estimating total energy via a graph-based function facilitates the use of graph algorithms to study material properties relevant to computing. In particular, we are interested in finding the *spin configuration* that produces the least amount of energy. This configuration is known as the *ground state* of the system. Barahona [2] proved that, for general spin systems, the *ground-state determination problem* (GSD) is NP-hard. Given that many physical systems have a natural ability to find least-energy states quickly, researchers are currently attempting to exploit this phenomenon to perform useful computation. At the atomic scale, energy minimization can be aided by quantum tunneling, which effectively reduces the number of local minima. Thus, GSD problems are of particular interest to quantum-information researchers because they are suitable candidates for evaluating the performance of *adiabatic quantum computers* (AQCs). Recently publicized AQCs employ an architecture based on Ising spin systems [13]. First, the system is configured to represent a given combinatorial problem, i.e., the spin interactions are carefully controlled rather than random as in statistical mechanics. The ground state is approximated via *quantum annealing* (the quantum analogue of thermal annealing), then read off as a bit sequence and interpreted as an answer to the problem. Bansal et al. [1] proposed an approximation algorithm for GSD on Ising spin lattices, *which essentially simulates these AQC architectures [8], and thus limits their potential for quantum speed-ups*. To approximate the least energy with $\epsilon$ accuracy, the algorithm from [1] requires runtime exponential in $1/\epsilon$, which is impractical. In contrast, we propose a branch-and-bound algorithm and a high-performance local search that quickly finds near-optimal energy values for arbitrary Ising topologies. Such techniques can be used to critically assess the performance of non-traditional computing devices based on energy minimization in spin-glasses and also to determine best implementation options. Our main contributions are:

- A branch-and-bound algorithm for solving GSD exactly on spin lattices with up to $100$ spins.
- A pass-based local search heuristic. Empirical results show that it scales better than other GSD algorithms and produces near-optimal solutions.
- A generalization of GSD to include many-spin couplings (*hyper-couplings*), and relevant algorithms.
- A study of how the omission of hyper-couplings impacts the success of AQC number factoring in recent NMR-based implementations.

The rest of the paper is structured as follows. In Section II we discuss the spin-glass model and the GSD problem. Section III presents our algorithms for finding ground states. Section IV reports empirical results. In Section V, we generalize the spin-glass model to include hyper-couplings and study their significance in number factoring.

## II. Background and Previous Work

In the Edwards-Anderson [6] model, spins are binary $\pm 1$ values, and the strengths of the atomic couplings are i.i.d. random variables according to some probability distribution. The most common distributions used are the Gaussian, and the $\pm 1$-bimodal distributions. Let $G_{ising} = (V, E)$ denote an Ising-model graph with $n$ vertices (spins). Each vertex $u \in V$ is denoted by spin value $S_u \in \{\pm 1\}$ and is assigned a magnetization weight $h_u$. For $u, v \in V$, we define $(u, v) \in E$ to be an edge representing a bond between two adjacent spins with assigned weight $J_{u,v}$ chosen randomly from the standard Gaussian distribution ($\mu = 0, \delta = 1$). Thus, we expect that half the bonds in the graph will be negative. Usually, the same positive- to-negative bond ratio is maintained when the $\pm 1$-bimodal distribution is used instead. The internal energy of the system for a particular configuration of spin values $\sigma = \{S_i\}$ is given by

$$E(\sigma) = -\sum_{(i,j)\in E}^{n} J_{i,j} S_i S_j - \sum_i h_i S_i \qquad (1)$$

where the first summation considers all pairs of adjacent spins. Putting together the energies of all spin configurations gives the *Hamiltonian* of the system. Thus, the ground state is given by $E_{gs} = \min(E(\sigma) \mid \forall\ \sigma \in \pi_n)$, where $\pi_n$ is the set of all possible $n$-spin configurations. Whether we are interested in the lowest-energy value or the $n$-spin configuration with such energy, $\mid \pi_n \mid = 2^n$ because each spin can take on one of two possible values. Since energy minimization is typically NP-hard, calculating the ground state exactly using exhaustive search is feasible only for small spin glasses. To provide a scalable way of finding ground states or approximating their energies, we need to employ heuristics.

The spin-glass topologies commonly considered in the literature are two- and three-dimensional lattices. To simulate infinite spin glasses, one requires *periodic boundary conditions*, which connect the spins lying on a dimensional boundary to the spins on the opposing boundary.

**Complexity of GSD**. Although most variations of GSD are known to be NP-hard [2], there are a few cases where the topology of the graph can be exploited to solve the problem in polynomial time. Bieche et al. [4] proved that the GSD problem on planar graphs can be solved in polynomial time by showing a reduction to the minimum-weight perfect matching (MWPM) problem. It follows from their work that GSD instances with zero magnetization ($h_i = 0$) and 0- or 1-periodic boundary conditions can be solved in $O(n^3)$ time. While MWPM is poly-time solvable, practical runtime and memory usage do not scale to very large instances. To overcome these limitations, the work in [10] describes a heuristic based on the MWPM reduction. Another special case is that of ferromagnetic ($J_{i,j} > 0$) GSD instances, which Barahona [3] reduced to ($s$-$t$)-min-cut or max-flow. Table I shows the spin-lattice properties that make GSD poly-time solvable and identifies the algorithms commonly used. Note that the number of dimensions, the number of boundary conditions and the presence of a magnetic field are the main factors in determining NP-hardness. More precisely, for lattices with more than two dimensions or with two boundary conditions, the graph is no longer planar and the reduction to MWPM breaks down. Compared to these techniques, our algorithms work on general instances of GSD, without limiting the structure of the underlying graph.

| Dims. | Boundaries | External magnet field | Bond weights | **NP-hard?** | Poly-time algorithm |
|---|---|---|---|---|---|
| 2 | 0 | No | ± | **No** | MWPM |
| 2 | 0 | Yes | ± | **Yes** | – |
| 2 | $\leq 2$ | Yes | + | **No** | Max-flow |
| 2 | 1 | No | ± | **No** | MWPM |
| 2 | 1 | Yes | ± | **Yes** | – |
| 2 | 2 | Yes/No | ± | **Yes** | – |
| $N > 2$ | $\leq N$ | Yes/No | ± | **Yes** | – |

TABLE I
SPIN-LATTICE PROPERTIES THAT MAKE GSD NP-HARD.

## III. GSD Algorithms

In order to better control trade-offs between runtime and solution quality obtained from heuristics, it is important to design algorithms that are guaranteed to find exact ground states on smaller instances. The exact solutions obtained from such instances can be used to debug and evaluate the performance of more scalable heuristics.

**Branch-and-bound (B&B)** algorithms consider incomplete or *partial* solutions, where only some variables are assigned values. Partial solutions are systematically constructed via a *branching* process that develops partial solutions that are deemed promising, i.e., those that may lead to the optimal solution. Partial solutions whose cost is too high, are "bounded away" or pruned.

Our branching process proceeds as follows. First, all spins are labeled as unassigned–their value can be set in the future to either $1$ or $-1$. The algorithm then calculates the lower bound $E_{lb}$ of Equation 1. It then selects a spin $i$ and branches on one of the possible values. In each branch, the incremental change in $E_{lb}$ caused by the assignment is recorded as follows. For each spin $j$ adjacent to $i$ that has already been assigned, increase (decrease) $E_{lb}$ by twice the amount of the positive (negative) bond connecting $i$ and $j$ if they have opposing (aligned) spin values,

$$E_{lb}^{\delta} = \begin{cases} 2\displaystyle\sum_{(i,j)\in E} J_{i,j} S_i S_j & \text{if } S_i \neq S_j \text{ and } J_{i,j} > 0, \\ -2\displaystyle\sum_{(i,j)\in E} J_{i,j} S_i S_j & \text{if } S_i = S_j \text{ and } J_{i,j} < 0 \end{cases} \qquad (2)$$

Similarly, the corresponding change due to the magnetization of the spin is also recorded,

$$E_{lb}^{\delta} = \begin{cases} 2h_i & \text{if } S_i = -1 \text{ and } h_i > 0, \\ -2h_i & \text{if } S_i = +1 \text{ and } h_i < 0 \end{cases} \qquad (3)$$

Once the spin is assigned, the algorithm branches out to another spin and performs the same procedure. When all spins have been assigned, $E_{lb}$ represents the energy of the spin configuration generated by the branching process. To

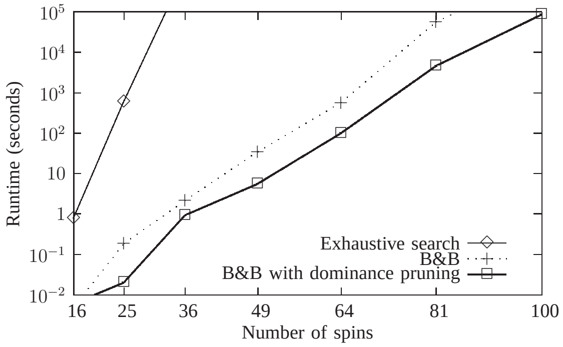


Fig. 1. Performance of B&B techniques on 2-D spin lattices.

continue searching the configuration space, the branching process backtracks to the last assigned spin, flips its value and updates $E_{lb}$. If both spin values have already been tried, then the algorithm continues backtracking while relabeling the spins as unassigned. Since each spin can take one of two values, this branching process generates a full binary search tree where the leaves correspond to all possible spin configurations in the Ising system.

Initially, we use a linear-time greedy approximation ($\tilde{E}$) as our bounding value. During the branching process, if the energy of the partial solution exceeds $\tilde{E}$, then we can safely prune this branch and backtrack without making any further assignments. The algorithm either tries the opposite spin value or backtracks again if both spin values have already been tried. If the search assigns all the spins in the graph and the corresponding minimal energy state is lower than $\tilde{E}$, then we set $\tilde{E}$ to this new energy value. After searching all promising branches, $\tilde{E}$ will assume the ground-state energy. This standard bounding technique alone improves the scalability of the branching process by an order of magnitude over exhaustive search (see Figure 1).

To further improve the scalability of our B&B algorithm, we designed a *prune-by-dominance* technique that consists of identifying partial solutions whose energy can be lowered by modifying the current spin assignment. For every assigned spin $s$, let $F_s$ be the set of spins adjacent to $s$ for which all neighboring spins have also been assigned, as illustrated in Figure 2. Note that $\mid F_s \mid \leq degree(s)$. The energy of the spins in $F_s$ is localized in the sense that it will not be affected by further spin assignments. Therefore, unpromising partial solutions can be identified by flipping the spins in $F_s$. If any flip lowers partial energy, the algorithm backtracks. Observe that when two branches are statistically unlikely to have equal partial cost (e.g., when couplings and magnetizations are random), the probability that the first branch dominates the second branch is approximately $1/2$. Let $c2^k$, $0 < c < 1$, be the partial solutions that require branching, then we expect to prune $c2^k/2 = c2^{k-1}$ of these. As Figure 1 shows, our pruning technique improved the scalability of B&B by 1-2 orders of magnitude, allowing it to solve 100-spin lattices in a day. However, since B&B takes exponential time in the worst case, it does not scale beyond 100 spins.

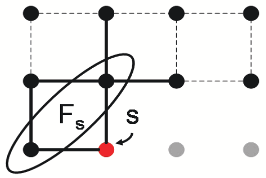


Fig. 2. Illustration of $s$ and $F_s$ on a small grid.

**Local search**. Due to its difficulty, GSD researchers have developed heuristic methods typically based on slow Monte Carlo simulations. However, because of the role that Ising models play in simulating real-world phenomena, faster techniques that work on non-grid topologies are desirable. Our local search is an iterative improvement algorithm that modifies the bipartition induced by an arbitrary spin configuration (positive spins are placed in one partition and negative spins in the other). The algorithm performs a sequence of incremental changes to the bipartition, organized as *passes*. These changes consist of *spin moves* that remove a particular spin from its current partition and places it in the opposite one. At the beginning of each pass, the energy differential (*gain*) of performing each possible move is calculated. A positive gain implies that the move decreases the overall energy while a negative gain increases it. During a pass, the move that produces the largest gain is selected and executed. The corresponding spin is then labeled as *locked*, i.e., it cannot be selected again in the current pass. The pass continues selecting and executing the best moves until all spins have been locked. At the end of the pass, we save the best-seen bipartition produced by the sequence of moves. This bipartition is then used as the starting solution of the next pass. The entire algorithm terminates when a pass fails to obtain an improvement in energy as shown in Figure 3. Note that, in the absence of positive-gain moves, a negative-gain move can be selected. Thus, a pass may accept a solution that is worse than the existing solution (hill-climbing). This helps to reduce the probability of getting trapped in local

```
Input: Ising spin-glass graph $G_{ising}$
Output: Approx. ground-state energy $E^*$
- - - - - - - - - - - - - - - - - - - - - - - - - -
Spin Partition $SP^* := RAND\_SP(G_{ising})$
while solution quality improves do
    Gains container $GC := COMPUTE\_GAINS()$
    Spin Partition $SP := SP^*$
    while $GC$ has unlocked spins do
        Move $m := SELECT\_BEST\_MOVE(GC)$
        $APPLY\_MOVE(SP, m)$
        $UPDATE\_GAINS(GC, m)$
        $LOCK\_SPIN(GC, m)$
        if energy decreased then $SP^* = SP$
    end while
end while
return $ENERGY(G_{ising}, SP^*)$
```

Fig. 3. Pass-based local search with hill-climbing

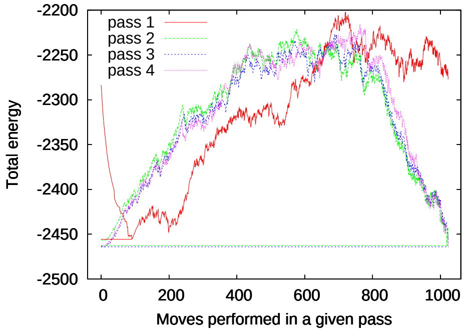


Fig. 4. Progress of our local search in terms of total energy during individual passes on a 1024-spin glass. The lowest energy state observed during each pass ($E_1 = -2456$, $E_2 = -2463$, $E_3 = -2465$, respectively) during a pass becomes the starting state of the next pass.

minima. Figure 4 illustrates the progress of our local search in terms of total energy during individual passes on a 1024-spin glass. The initial random solution used is generated in linear time. To increase the quality of solutions and the probability of finding the exact ground state, we repeat the algorithm using multiple random initial solutions and selecting the best result. The relationship between the number of random starts and solution quality is explored in Section IV.

These gain updates are performed efficiently using a custom heap-based data structure. The data structure consists of two arrays. The first array implements a traditional binary heap while the second array allows quick access to the heap-array element that contains the gain-update value of a particular spin. To perform gain updates, we can access the specific value in $O(1)$ time, update the value, and perform the necessary swaps to maintain heap order. Since only $\log n$ swaps are required in the worst case (where $n$ is the number of spins), our data structure updates gain values much faster than naive implementations that require scanning the entire set of $n$ gain values. Since $n$ moves are performed during a pass, and only a constant number of passes are required, the runtime of our local search is $O(n \log n)$.

**Comparison with Fidducia-Matheyses.** The concept behind our local search was first proposed by Fiduccia and Mattheyses (FM) [5] in the context of VLSI netlist partitioning. However, the two techniques pursue very different objectives and observe different constraints. They also require different algorithmic details and different data structures. For example, in the context of GSD problems, it is not necessary to maintain balance constraints–a common requirement for most FM-based VLSI graph partitioning algorithms. This makes the problem easier since balanced bipartitioning is NP-hard. On the other hand, our spin-partitioning technique must handle real-valued positive and negative edges, which makes the problem hard again (only non-negative integer edge weights are used in VLSI partitioning). In VLSI netlist partitioning, the objective is to minimize the cut of the bipartition, which only considers the weights of the edges. Our local search minimizes the overall energy function (see Equation 1), which takes into account both the edge weights (bond strength) and vertex weights (spin magnetization).

## IV. Empirical Validation

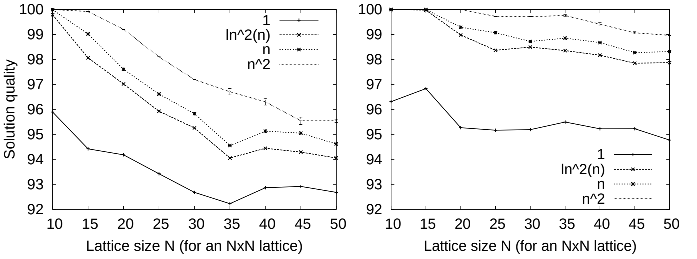
We evaluated single-threaded implementations of our algorithms on a conventional Linux server, although our local search is trivial to parallelize to a multicore system or a distributed cluster. For $15 \times 15$ spin lattices our local search finds exact ground states in $95\%$ of independent random starts (exact solutions were obtained from [15]), otherwise solutions are $5\%$ sub-optimal on average. Figure 5(a) compares the average solution quality of local search for 2-D spin glasses with Gaussian-distributed couplings and $h_i = 0$ (instances with $h_i \neq 0$ are not allowed in [15]). For each instance we considered four different levels of effort with an increasing number of independent random starts. To obtain the average solution quality we computed $1000$ output samples using $1$, $\ln^2 n$ and $n$ random starts (where $n$ is the total number of spins) per instance. For $n^2$ random starts, we used fewer output samples and provide confidence intervals. As expected, the solution quality improves as the number of random starts increases. When at least $\ln^2 n$ random starts are used, our heuristic produces high-quality solutions ($> 95\%$) for five of the benchmarks while its runtime does not exceed $17$ seconds for the largest benchmark ($2500$ spins). Note that the expected solution quality slowly decreases for larger instances. Figure 5(b) shows similar results for benchmarks with $\pm 1$-bimodal coupling distributions, but solutions are closer to optimal. For all but one of the benchmarks, our heuristic requires only a single random start to find high-quality solutions.

**Local optimality**. We verified that the configurations returned by our heuristic cannot be improved by modifying a small number of spins. We used our B&B to find optimal configurations of groups of $25$-$49$ adjacent spins within larger configurations. In our experiments, the solutions produced by our heuristic were never improved by this technique.

**Runtime**. Our heuristic scales to a million spins and empirical runtimes closely fit $n \log(n)$. We compared the runtime of our local search against that of the MWPM-based heuristic proposed in [10]. The solution quality of this heuristic depends on a particular choice of parameters and does not work on Gaussian-distributed instances. In contrast, our heuristic does not have such dependencies and works on all instances. Table IV in [10] shows the runtime and solution quality of the MWPM-based heuristic $\pm 1$ grid graphs of size $164 \times 164$. This heuristic takes $59$s on average producing the optimal value $61\%$ of the time. By comparison, a single run of our heuristic on a comparable benchmark takes about $8.5$ seconds. Thus, we can perform $7$ random starts in the same period of time. However, we could not compare the solution quality as we did not have access to the same benchmarks.

## V. GSD for hypergraphs

As discussed in Section I, recent AQC architectures [13] are based on spin glasses. Implementations described by D-Wave Systems use non-planar topologies, and recent experiments in [11] demonstrate direct coupling of more than two spins. Hence, we extend the conventional spin glass model to use



(a) Gaussian-distributed couplings (b) Bimodal-distributed couplings

Fig. 5. Expected solution quality (100% corresponds to the exact solution obtained from [15]) produced by using different numbers of random starts on 2-D spin lattices. However, our techniques are not limited to lattices where optimal solutions are found.

*hyper-couplings* that connect a set of at least two spins. The new energy function is given by

$$E(\sigma) = -\sum_{e\in E} J_e \prod_{i\in e} S_i - \sum_{i\in V} h_i S_i \qquad (4)$$

With minor modifications, our algorithms can optimize this function allowing us to study a greater variety of physical systems and solve a wider range of combinatorial problems.

**Number-factoring as optimization**. Integer factorization is equivalent to the optimization of $f(x, y) = (N - xy)^2$, where $N$ is the odd integer we wish to factor and $x$, $y$ are odd positive integers $< N$. The minimum of $f(x, y)$ is reached when $x$ and $y$ are factors of $N$. To solve this optimization problem using GSD, we construct an Ising system whose ground state corresponds to the minimum of $f(x, y)$. Since spins are $\pm 1$ binary variables, we reformulate $f(x, y)$ in terms of binary digits such that $x = \sum_{i=1}^{n_x-1} 2^i x_i$ and $y = \sum_{i=1}^{n_y-1} 2^i y_i$. Since $x_0 = y_0 = 1$, the total number of spins is $(n_x-1)+(n_y-1) = \mid V \mid$. Setting $S_i = 2x_i - 1$ and $S_{n_x+i} = 2y_i - 1$,

$$f(x,y) = \Bigg[N - \left(2^{n_x-1}\frac{1-S_{n-1}}{2} + ... + 2\frac{1-S_{n-n_x}}{2} + 1\right) \left(2^{n_y-1}\frac{1-S_{n-n_x-1}}{2} + ... + 2\frac{1-S_1}{2} + 1\right)\Bigg]^2 \qquad (5)$$

The magnetization weights and coupling strengths are given by the product expansion of the above equation.

**Computational experiments**. B&B factors up to 21-bit numbers in about 5 minutes. While leading-edge number-factoring algorithms can do better, the results confirm that B&B is general enough to work on hyper-Ising models. We tested our GSD heuristic by factoring specific numbers. In some runs, this technique produces the factors of $N \pm 2$ or other incorrect numbers. Therefore, many independent random runs may be required to factor a given number $N$, which is also typical behavior in AQC. Figure 6 shows the output probability distribution of factoring the number $612171$, which has several factors and is therefore easy to factor. The probability of success drops sharply for semi-primes, e.g., the probability of factoring $580003$ in one attempt is only $.005$. Our implementation factored the semi-prime $10185081163 = 100511 \times 101333$ using about $15,000$ random starts in $13$ seconds. While further optimizations may significantly improve runtime, present results provide a computational baseline for performance evaluation of novel computing devices that solve hard problems via energy minimization. Viable AQC devices would need to improve on our results by producing output distributions with greater probability of factoring the correct number.

**Significance of hyper-couplings**. Recently, Peng et al. [13] implemented an AQC number-factoring algorithm in NMR technology with a Hamiltonian that ignores hyper-couplings. By simulating the functionality of the algorithm from [13], we replicated their experiment and explored factorization of larger numbers. As in [13], the number $21$ was successfully factored even in the absence of hyper-couplings. However, our experiments show that, in general, omitting hyper-couplings alters the ground states so that correct factors cannot be found, e.g., for $35$ and $91$. Instead, some other numbers are factored such as $33$ and $95$. Although hyper-couplings have smaller individual weights than two-spin couplings, the number of hyper-couplings scales as $O(n^4)$, and their total weight eventually dominates $f(x, y)$ for larger $N$. Control of three-spin hyper-couplings has recently been demonstrated [11], but only for adjacent particles, which would be insufficient for number-factoring applications. Furthermore, Equation 5 still requires four-spin hyper-couplings which, as current research suggests, are difficult to realize experimentally.

**Avoiding hyper-couplings via ancilla spins**. The work in [14] shows a method for expanding Equation 5 that avoids the use of hyper-couplings at the cost of a quadratic increase in spins. The new Hamiltonian is based on the factorization equations that are generated by decomposing long-hand binary multiplication [14]. Equation 5 is modified by introducing ancilla binary variables. Let $N = m_1 m_2 ... m_n$. Let $x = x_1 x_2 ... x_k$ and $y = y_1 y_2 ... y_{n-k}$. Then let $p_{i,j}$ denote the sum of pairwise products between the binary variables in Equation 5 and let $c_{i,j}$ denote carry variables

$$f(x,y) = \sum_{i=1}^{k}\sum_{j=1}^{n-k} \Big[2\left(x_i/2 + y_j/2 + p_{i,j} + c_{i,j} - p_{i+1,j-1} - 2c_{i-1,j} - 1/4\right)^2 - 1/8\Big]$$

The bits of $N$ are linked to the bits of $x$ and $y$ by a series of equalities such as $p_{i,0} = m_i$ and $p_{k+1,j-1} = m_{k+j}$ (see [14] for details). $f(x, y)$ now computes a penalty function for violating the factorization equations. This penalty is minimized subject to the fixed values of $m_1 ... m_n$. The corresponding spin system is constructed in the same manner as before except that only two-spin couplings are generated. Unlike the (unconstrained) formulation with direct hyper-couplings, this formulation includes spins with fixed logic values and thus requires additional technology support (e.g., through optical pumping of trapped ions). We used our algorithms as simulators to compare the direct use of arbitrary hyper-couplings (assuming technological feasability) to the expansion from [14]. Figure 6 compares output probability distributions generated when factoring $51$. The technique from [14] produces a flatter distribution with a wider range. This is because the

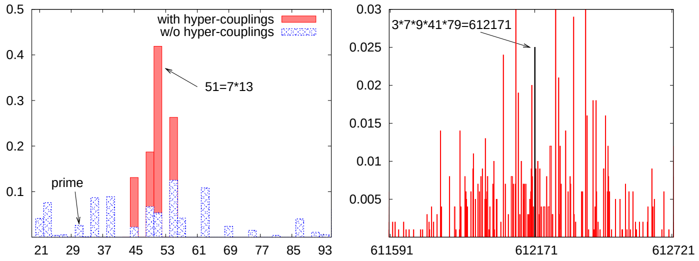


Fig. 6. Output prob. distributions for factoring $51$ and $612171$ using the techniques from [13] and [14] as simulated by our algorithms.

solution space is more complex and includes configurations where the ancillary spins are inconsistent (e.g., carry spin $c_{i,j} = 0$, but the partial product spin $p_{l,k} = c_{i,j} * 1 = 1$). Thus, this technique sometimes returns trivial decompositions of prime numbers (e.g., 31), whereas direct use of hyper-couplings always results in proper factorizations. In summary, our experiments suggest that the expansion from [14] is not computationally competitive with direct use of hyper-couplings.

## VI. Conclusions

The problem of finding the least-energy state of a spin system (GSD) has played a central role in statistical physics for decades [7, Chapter 3]. In modern EDA research, high-performance GSD algorithms can $(i)$ link spin-level descriptions of non-traditional compute substrates to energy-based statistical macro models, and $(ii)$ help evaluate architectural alternatives in adiabatic computing. Recent work in [12] concludes that quantum annealing loses to classical simulated annealing in a head-to-head comparison, but can be improved by making different architectural choices. Although our algorithms do not simulate quantum optimization directly, they allow one to study problem reductions and identify potential obstacles to successful optimization. The proposed B&B algorithm can find ground states for general 2-D spin lattices with up to $100$ spins in $24$ hours. For GSD instances with $1,000,000$ spins, our local search heuristic obtains approximate solutions in $< 4$ hours. It provides a scalability advantage over conventional Monte Carlo methods and is not limited to special classes of GSD instances. When our heuristic does not find a ground state, it usually approximates the least energy within $5\%$.

The algorithms in Spinto allow one to study the significance of specific aspects of recent proposals for non-traditional computing. In particular, we replicated a recently published empirical result in AQC-based number factoring [13], where the omission of spin-spin hyper-couplings did not undermine overall results. However, we have shown that, in general, omitting hyper-couplings will produce incorrect results. Furthermore, we demonstrated that techniques avoiding the use of hyper-couplings [14] blur the output probability distribution, hamper finding correct factors, and will require many more repeated attempts to achieve success.

**Acknowledgement.** This material is based on research sponsored by the Air Force Research Laboratory under agreement number FA8750-10-2-0016. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

## References

[1] N. Bansal et al., “Classical approximation schemes for the ground–state energy of quantum and classical Ising spin Hamiltonians on planar graphs,” *Quant. Inf. Comp.*, 2009(7).

[2] F. Barahona, “On the computational complexity of the Ising spin glass models,” *J. Phys. A*, 1982, pp. 3241–3253.

[3] F. Barahona, “Ground–state magnetization of Ising spin glasses,” *Phys. Rev. B*, 1994, 49(18), pp. 12864–12867.

[4] L. Bieche, et al. “On the ground states of the frustration model of a spin glass by a matching method of graph theory,” *J. Phys. A: Math. Gen.*, 13 (1980) pp. 2576.

[5] C. M. Fiduccia and R. M. Mattheyses, “Linear time heuristic for improving network partitions”, *DAC*, 1982, pp. 175-181.

[6] S. F. Edwards, P. W. Anderson, *J. Phys. F*, 1975, pp. 965.

[7] A. K. Hartmann, H. Rieger, “New optimization algorithms in physics”, *Wiley-vch*, 2004.

[8] W. M. Kaminsky et al., “Scalable superconducting architecture for adiabatic quantum computation,” 2004, arXiv:quant-ph/0403090v2.

[9] R. Oliveira, B. M. Terhal,“The complexity of quantum spin systems on a two-dimensional square lattice,” *Quant. Inf. Comp.*, 8(10), pp. 900–924(2008).

[10] G. Pardella, F. Liers, “Exact ground states of large two-dimensional planar Ising spin glasses,” *Phys. Rev. E 78*, 056705 (2008).

[11] K. Kim et al.,“Entanglement and tunable spin-spin couplings between trapped ions using multiple transverse modes,” 2009, arXiv:quant-ph/0905.0225v1.

[12] D. A. Battaglia, G. E. Santoro, E. Tosatti, “Optimization by quantum annealing: lessons from hard satisfiability problems”, *Phys Rev E 71* 066707 (2005).

[13] X. Peng et al., “A quantum adiabatic algorithm for factorization and its experimental implementation,” Phys. Rev. Lett. 101, 220405 (2008).

[14] G. Schaller, R. Schutzhold, “The role of symmetries in adiabatic quantum algorithms,” 2009, arXiv:quant-ph/0708.1882v2.

[15] http://www.informatik.uni-koeln.de/ls_juenger/research/sgs/