An Ising Framework for Constrained Clustering on Special Purpose Hardware

Eldan Cohen*, Arik Senderovich†, and J. Christopher Beck*

*Department of Mechanical & Industrial Engineering and †Faculty of Information
University of Toronto, Toronto, Canada
{ecohen, sariks, jcb}@mie.utoronto.ca

Abstract. The recent emergence of novel hardware platforms, such as quantum computers and Digital/CMOS annealers, capable of solving combinatorial optimization problems has spurred interest in formulating key problems as Ising models, a mathematical abstraction shared by a number of these platforms. In this work, we focus on constrained clustering, a semi-supervised learning task that involves using limited amounts of labelled data, formulated as constraints, to improve clustering accuracy. We present an Ising modeling framework that is flexible enough to support various types of constraints and we instantiate the framework with two common types of constraints: pairwise instance-level and partition-level. We study the proposed framework, both theoretically and empirically, and demonstrate how constrained clustering problems can be solved on a specialized CMOS annealer. Empirical evaluation across eight benchmark sets shows that our framework outperforms the state-of-the-art heuristic algorithms and that, unlike those algorithms, it can solve problems that involve combinations of constraint types. We also show that our framework provides high quality solutions orders of magnitudes more quickly than a recent constraint programming approach, making it suitable for mainstream data mining tasks.

1 Introduction

Recent years have seen the emergence of novel computational platforms, including adiabatic and gate-based quantum computers, Digital/CMOS annealers, and neuromorphic computers (for a review see [8]). These machines represent a challenge and opportunity to AI and OR researchers: how can specialized models of computation as embodied by the new hardware be harnessed to better solve AI/OR problems. Several new hardware platforms have adopted Ising models [19] as their mathematical formulation and, consequently, a number of existing problems have been formulated as Ising models, including clustering [22], community detection [34], and partitioning, covering, and satisfiability [26].

Constrained clustering is a semi-supervised learning task that exploits small amounts of labelled data, provided in the form of constraints, to improve clustering performance [35]. In the past two decades, this topic has received significant attention and algorithms that support different types of constraints have been proposed [6, 29, 24]. As finding an optimal solution to the (semi-supervised) clustering problem is an NP-hard problem [27], the commonly used algorithms rely on heuristic methods that quickly converge to a local optimum.

In a recent work, Kumar et al. [22] presented an Ising model for unsupervised clustering and observed mixed results using a quantum annealer. However, formulating constrained clustering problems as Ising models and solving them in hardware has not been studied. In this work, we introduce and analyze a novel Ising modeling framework for semi-supervised clustering that supports the combination of different types of constraints and we instantiate it with pairwise instance-level and partition-level constraints. We demonstrate the performance on the Fujitsu Digital Annealer [28, 33], and discuss several hardware-related considerations when embedding our framework on this hardware.

Our main contributions are summarized as follows:

- We introduce an Ising framework for constrained clustering with pairwise and partition-level constraints that can be solved on a variety of novel hardware platforms.
- We demonstrate the performance of our framework on a specialized CMOS annealer and show that it outperforms the state-of-the-art heuristic methods for constrained clustering and produces approximately equal or better solutions compared to a constraint programming model in a small fraction of the runtime (i.e., a two orders of magnitude speed-up).
- We show that the framework can seamlessly solve semi-supervised clustering problems with both pairwise and partition constraints, problems that cannot be solved by the existing heuristic techniques.
- We discuss some of the challenges in embedding Ising models onto quantum and quantum-inspired hardware.

2 Background

Let $X = \{x_i\}_{i=1}^{n}$ be the set of n data points with x_i being a finite-sized feature vector and K be the number of clusters ($K<n$). Combinatorial clustering algorithms attempt to find a partition of X into K disjoint subsets, $S = S_1 \cup \cdots \cup S_K$, that minimizes a chosen objective function, typically the total within-cluster scatter [32] based on pairwise dissimilarities, $d(x_i, x_j)$. When the dissimilarity is represented by the squared Euclidean distance the objective is:

$$\min \sum_{k=1}^{K} \sum_{\substack{i<j: \\ x_i, x_j \in S_k}} d(x_i, x_j) = \sum_{k=1}^{K} \sum_{\substack{i<j: \\ x_i, x_j \in S_k}} \|x_i - x_j\|^2. \tag{1}$$

In the Euclidean case, another commonly used objective function is the sum of squared errors [18],

$$\min \sum_{k=1}^{K} \sum_{x_i \in S_k} \|x_i - \mu_k\|^2, \tag{2}$$

where μ_k is the mean vector of the points in cluster k.

2.1 Constrained Clustering

In a semi-supervised setting, we assume some amount of labelled data in the form of constraints. Constrained clustering is the problem of finding a partition that satisfies the provided constraints [35]. First, we consider two pairwise constraints: must-link (ML) and cannot-link (CL) [4]. ML constraints are defined by a set, $\mathcal{M}$, of pairs of points that must be assigned to the same cluster, $(x_i, x_j) \in \mathcal{M} \Rightarrow s(x_i) = s(x_j)$, where $s(x_i)$ denotes the cluster that x_i is assigned to, $s(x_i) = k \iff x_i \in S_k$. CL constraints are defined by a set, $\mathcal{C}$, of pairs of points that must be assigned to different clusters, $(x_i, x_j) \in \mathcal{C} \Rightarrow s(x_i) \neq s(x_j)$.

Bilenko et al. [6] proposed the *Pairwise Constrained K-Means (PCK-Means)* problem that incorporates the constraints in the objective function:

$$\begin{aligned} \min \sum_{k=1}^{K} \sum_{x_i \in S_k} \|x_i - \mu_j\|^2 + \sum_{(x_i,x_j)\in\mathcal{M}} w_{i,j} \mathbb{1}[s(x_i) \neq s(x_j)] \\ + \sum_{(x_i,x_j)\in\mathcal{C}} \overline{w}_{i,j} \mathbb{1}[s(x_i) = s(x_j)] \end{aligned} \tag{3}$$

where $\mathbb{1}[true] = 1$ and $\mathbb{1}[false] = 0$. PCK-Means is solved using a greedy iterative algorithm, adapted from the K-Means algorithm [25]. Note that Eq. (3) allows violation of ML and CL constraints depending on the weights $w_{i,j}$ and $\overline{w}_{i,j}$ that correspond to the confidence in the external information [23]. *Metric PCK-Means (MPCK-Means)* [6] is a combination of PCK-Means with distance-metric learning [36] that outperforms PCK-Means [9].

Other well-known approaches include *Constrained Vector Quantization Error (CVQE)* [13] that augments the clustering objective to account for constraint violations, but uses the distances between the centroids to compute the violation costs, and *linear-time CVQE (LCVQE)* [29] that computes the violation costs based on the distances between objects and centroids. LCVQE was found to be competitive in terms of accuracy with CVQE while violating fewer constraints [9].

We also consider partition-level (PL) constraints, where some points have predefined cluster labels. Formally, assuming an arbitrary labeling of clusters k, $X_k \subseteq X$ denotes the set of points that must be assigned to cluster k. For example, in clustering of patients into two cancer risk categories, X_1 (X_2) is the set of patients known to have low (high) risk of having cancer.

To handle PL constraints, Liu et al. [24] proposed the *Partition-Level Constrained Clustering (PLCC)* problem that uses the following objective:

$$\min \sum_{k=1}^{K} \sum_{x_i \in S_k} \|d_i^{(1)} - m_k^{(1)}\|^2 + \Lambda \mathbb{1}[d_i \in P] \|d_i^{(2)} - m_k^{(2)}\|^2 \tag{4}$$

where the first term is the squared distance from centroid and the second term is the constraint violation weighted by Λ. *PLCC* is solved using a K-Means-like algorithm.

Several works have applied model-based exact techniques to constrained clustering, including constraint programming [10–12] and integer linear programming [2]. In a recent work, Dao et al. [12] proposed a constraint programming (CP) approach for constrained clustering that minimizes within-cluster pairwise dissimilarity (Eq. (1)) using a dedicated global constraint. In an earlier work, they showed that a similar CP approach for minimizing sum of squared errors outperforms integer programming [11]. Although exact techniques are able to find and prove optimal solutions, they are often several orders of magnitude slower than heuristic techniques and for large problems can be intractable. Furthermore, they do not return a solution in case of contradictory constraints.

2.2 Ising Models

Ising models are graphical models that comprise a set of nodes $\mathcal{N}$ representing *spin* variables, $\sigma_i \in \{-1, 1\}, i \in \mathcal{N}$ and a set of edges $\mathcal{E}$ representing *interactions* between spin variables, $(i, j) \in \mathcal{E}$. The problem is parameterized by the *biases* h_i and the *couplers* $J_{i,j}$. The objective is to minimize the *energy* of the model given by the Hamiltonian:

$$E(\sigma) = \sum_{(i,j)\in\mathcal{E}} J_{i,j}\sigma_i\sigma_j + \sum_{i\in\mathcal{N}} h_i\sigma_i. \tag{5}$$

Quadratic unconstrained binary optimization (QUBO) models are equivalent representations used to model problems with *binary* decision variables. Specifically, a QUBO model has n decision variables, $q_i \in \{0, 1\}, i \in [1..n]$, with corresponding *biases*, c_i, and *couplers*, $c_{i,j}$. The objective of the QUBO is to minimize the following quadratic function:

$$E(q) = \sum_{i=1}^{n} c_i q_i + \sum_{i<j} c_{i,j} q_i q_j. \tag{6}$$

QUBO models can be converted to Ising models by setting $\sigma_i = 2q_i - 1$ [5] and thus we refer to them as Ising models.

2.3 Unsupervised Clustering with Ising Models

Kumar et al. [22] presented a QUBO model for unsupervised clustering,

$$E(q) = \sum_{i<j} c_{i,j} \sum_{k=1}^{K} q_k^i q_k^j + \sum_{i=1}^{n} \lambda_i \phi_i. \tag{7}$$

The first term in the objective is the within-cluster all-pairs dissimilarity. The cluster assignment for each data point is represented using one-hot encoding, i.e., K binary variables q_k^i such that $q_k^i{=}1 \iff x_i{\in}S_k$. Since each point is assigned to exactly one cluster, the QUBO model includes a quadratic penalty term to ensure the one-hot encoding holds:

$$\phi_i = \left(\sum_{k=1}^{K} q_k^i - 1 \right)^2 . \tag{8}$$

If x_i is assigned to exactly one cluster $\phi_i = 0$, otherwise $\phi_i \geq 1$ and the objective is penalized by $\lambda_i \phi_i$.

Kumar et al. [22] could only fit very small instances on a quantum annealer (up to 40 points) and used classical solver for larger instances. Their results were, at best, competitive with the K-Means heuristic in terms of solution quality.

3 A Framework for Constrained Clustering

We start by formulating the semi-supervised constrained clustering problem as a constrained optimization problem (COP). Given a problem instance defined by $\langle X, K, \mathcal{M}, \mathcal{C}, \{X_k\}_1^K \rangle$, we wish to find a partition, $S = S_1 \cup \cdots \cup S_K$, that minimizes the objective in Eq. (1) while satisfying the constraints:

$$\begin{aligned} \min_S \quad & \sum_{k=1}^{K} \sum_{\substack{i<j: \\ x_i, x_j \in S_k}} \|x_i - x_j\|^2 \\ \text{s.t.} \quad & s(x_i) = s(x_j), \qquad \forall (x_i, x_j) \in \mathcal{M} \\ & s(x_i) \neq s(x_j), \qquad \forall (x_i, x_j) \in \mathcal{C} \\ & s(x_i) = k, \qquad\qquad \forall k \in K, \forall x_i \in X_k . \end{aligned} \tag{9}$$

3.1 A QUBO Model for Constrained Clustering

We modify the unsupervised clustering model (Eq. (7)) to include clustering constraints. Specifically, we introduce the pairwise and partition-level constraints as quadratic penalty terms in the energy function:

$$\begin{aligned} E(q) = & \sum_{i<j} c_{i,j} \sum_{k=1}^{K} q_k^i q_k^j + \sum_{i=1}^{n} \lambda_i \phi_i \ + \sum_{\substack{i<j: \\ (x_i, x_j) \in \mathcal{M}}} w_{i,j}^{\mathcal{M}} \psi_{(i,j)}^{M} + \\ & \sum_{\substack{i<j: \\ (x_i, x_j) \in \mathcal{C}}} w_{i,j}^{\mathcal{C}} \psi_{(i,j)}^{C} \ + \sum_{k=1}^{K} \sum_{i: x_i \in X_k} w_{i,k}^{P} \psi_{(i,k)}^{P} . \end{aligned} \tag{10}$$

The cost function is $c_{i,j} = \|x_i - x_j\|^2$ and the terms $\lambda_i \phi_i$ enforce the one-hot encoding (Eq. (8)). The terms $w_{i,j}^{\mathcal{M}} \psi_{(i,j)}^{M}$ enforce must-link constraints by penalizing the energy function if x_i and x_j are assigned to different clusters,

$$\psi_{(i,j)}^{M} = \sum_{k=1}^{K} (q_k^i - q_k^j)^2 , \tag{11}$$

with $(q_k^i - q_k^j)^2$ being quadratic terms equal to one if $q_k^i \neq q_k^j$ and zero if $q_k^i = q_k^j$.[1]

The terms $w_{i,j}^{\mathcal{C}} \psi_{(i,j)}^C$ enforce the cannot-link constraints by penalizing the energy function if x_i and x_j are in the same cluster, i.e., there exists k such that $q_k^i = 1$ and $q_k^j = 1$,

$$\psi_{(i,j)}^C = \sum_{k=1}^{K} q_k^i q_k^j. \tag{12}$$

The terms $w_{i,k}^P \psi_{(i,k)}^P$ enforce the partition-level constraints by penalizing the energy function for assigning a data point $x_i \in X_k$ in a cluster $m \neq k$,

$$\psi_{(i,k)}^P = \sum_{\substack{m=1,\\ m\neq k}}^{K} q_m^i. \tag{13}$$

Once we obtain a solution to the QUBO in Eq. (10), each point x_i is represented by K bits $q_k^i, k{\in}[1..K]$ where $q_k^i = 1$ if and only if x_i is in cluster k. We can extract the cluster for each point using the following function:

$$z_i(q) = \arg\max_{k\in[1..K]} q_k^i. \tag{14}$$

If the one-hot encoding constraint is satisfied, z_i is bijective and therefore the partition can be obtained as follows:

$$x_i \in S_k \iff z_i(q) = k. \tag{15}$$

3.2 Choosing the Weights

Given Eq. (10), we must choose weights for the penalty terms to control the constraint violation. In most practical cases, the one-hot encoding is a hard constraint that we do not want violated. However, depending on the confidence we have in each of the constraints, we may be willing to violate some of these constraints in favor of satisfying others.

We consider the case in which our constraints come from a trusted source and we wish to find a partition that satisfies all constraints. Setting the weights for all penalty terms to be $n\tilde{d}$, where $\tilde{d} = \max c_{i,j}$, guarantees that the optimal solution to the QUBO model in Eq. (10) is an optimal solution for the COP in Eq. (9).

Theorem 1 *Consider a constrained clustering problem defined by $\langle X, K, \mathcal{M}, \mathcal{C}, \{X_k\}_1^K \rangle$, such that the COP in Eq. (9) is satisfiable. Let $E(q)$ be the energy function in our QUBO model (Eq. (10)), with the following weights for the penalty terms $\lambda_i = w_{i,j}^{\mathcal{M}} = w_{i,j}^{\mathcal{C}} = w_{i,k}^P = n\tilde{d}$. Let $\bar{q}$ be an optimal solution to our QUBO model. Then the corresponding partition $\bar{S}$, $x_i \in \bar{S}_k \iff z_i(q) = k$, is an optimal solution to the COP in Eq. (9).*[2]

[1] If the one-hot encoding constraint is satisfied, violating a must-link constraint will apply two penalty terms, one for each of the two clusters of the data points.

[2] All proofs appear in tidel.mie.utoronto.ca/pubs/constrained-clustering-proofs.pdf.

3.3 An Efficient Encoding for $K = 2$

In the special case of $K = 2$, we can use an encoding that only requires n variables, rather than Kn variables:[3]

$$\begin{aligned} E_B(p) = &\sum_{i<j} c_{i,j}(p^i{+}p^j{-}1)^2 + \sum_{\substack{i<j:\\(x_i,x_j)\in\mathcal{M}}} \hat{w}^{\mathcal{M}}_{i,j}\sigma^M_{(i,j)} + \\ &\sum_{\substack{i<j:\\(x_i,x_j)\in\mathcal{C}}} \hat{w}^{\mathcal{C}}_{i,j}\sigma^C_{(i,j)} + \sum_{k=1}^{K}\sum_{i:x_i\in X_k} \hat{w}^P_{i,k}\sigma^P_{(i,k)}. \end{aligned} \tag{16}$$

The variables p^i represent the partition: x_i is in the first cluster if $p^i = 0$ and in the second cluster otherwise. The terms $\sigma^M_{(i,j)} = (p^i - p^j)^2$ enforce the must-link constraints, the terms $\sigma^C_{(i,j)} = (p^i + p^j - 1)^2$ enforce the cannot-link constraints, and the terms $\sigma^P_{(i,k)} = [p^i - (k-1)]^2$ enforce the partition-level constraints.

Theorem 2 shows that the equivalence between the efficient encoding and the general model for $K = 2$. The bound in Theorem 1 is therefore applicable for this model.

Theorem 2 *Consider a constrained clustering problem defined by $\langle X, K, \mathcal{M}, \mathcal{C}, \{X_k\}_1^K\rangle$ such that $K = 2$. Let q^i_k be an assignment of variable for the K-clustering model in Eq. (10). We set $\hat{w}^M_{i,j} = 2w^M_{i,j}$, $\hat{w}^C_{i,j} = w^C_{i,j}$ and $\hat{w}^P_{i,j} = w^P_{i,j}$. If the one-hot encoding constraint is satisfied (i.e., $\phi_i = 0$ in Eq. (10)), then $E(q) = E_B(p)$ where p^i is equal to zero if $q^i_1 = 1$ and equal to one if $q^i_2 = 1$.*

4 Constrained Clustering on the Fujitsu Digital Annealer

The Fujitsu Digital Annealer (DA) is recent CMOS hardware designed for Ising optimization problems formulated as a QUBO [28, 33]. We use the first generation of the DA that is capable of representing problems with up to 1024 variables with 16-bit precision for the couplers and 26-bit precision for the biases.

The DA algorithm is based on simulated annealing [21], however it takes advantage of the massive parallelization provided by the custom CMOS hardware [1]. Furthermore, it has several key differences compared to simulated annealing:

- It starts every run from the same arbitrary state to reduce computational effort.
- It uses a *parallel-trial* scheme in which each Monte Carlo step considers all possible one-bit flips, in parallel. If more than one flip is accepted, one of accepted flips is chosen uniformly at random.
- It uses *dynamic offset* to increase the energy of a state in order to escape local minima.

[3] Kumar et al. [22] presented a model for unsupervised clustering with n variables for K=2. Their model uses spin-glass variables and does not optimize the energy function in Eq. (10).

4.1 Embedding Problems on the DA

When solving constrained clustering problems on the DA we have to make some practical representation and configuration choices. Due to the precision limit, we need to embed the couplers and biases on a scale with limited granularity. We therefore make the following implementation choices:

1. The distances $d(x_i, x_j)$ are normalized in the discrete range of $[0, 150]$.
2. The chosen weights cannot be arbitrarily high and the bound in Theorem 1 cannot be met. Instead we use the highest supported value for λ, the weight that enforces the one-hot encoding.
3. The bound in Theorem 1 guarantees that all constraints are satisfied if the problem is solved to optimality. In practice, the DA does not necessarily solve problems to optimality and instead terminates after a specified time limit. To avoid cases where the DA violates a one-hot encoding constraint in favor of satisfying a clustering constraint, we empirically find that it is better to use a lower weight for the penalty terms of the clustering constraints. In our experiments, we used a ratio of 1:4, $w^{\mathcal{M}} = w^{\mathcal{C}} = w^{\mathcal{P}} = \frac{1}{4}\lambda$.

The optimization parameters that represent the temperature schedule are tuned once per data set based solely on the obtained objective value (we do not use the true labels).

Unlike K-Means-based algorithms that run until convergence, our method runs for a given time limit and returns the best solution encountered. We therefore need to define a time limit to use in the evaluation of our approach. Considering the run time of heuristic techniques can vary significantly (for example, Liu and Fu [23] found LCVQE average run time varies between 0.01 to 76.73 seconds across different data sets) and the needs of practical applications, we arbitrarily choose 5 seconds as a time limit for each execution of our model (see Section 5.6 for further discussion).

5 Empirical Evaluation

We perform an empirical evaluation of our method across eight benchmark data sets. As the commonly used methods only support one type of constraint (pairwise or partition-level), we first compare performance on problems with one constraint type. Then, we evaluate our method on problems that involve *both* pairwise and partition-level constraints. To demonstrate the advantages of using special purpose hardware for combinatorial optimization, we compare our method to constraint programming [12] and two CPU solvers for Ising models.

5.1 Data sets

We run experiments on eight data sets: *Breast Cancer, Ionosphere, Pima, Sonar, Seeds, Optdigits, Letters* [15], and *Protein* [36]. *Optdigits-389* is a randomly sampled subset of the UCI handwritten digits data set containing only the digits

Table 1. Description of data sets

Data set	Instances	Features	Classes	CV
Breast cancer	683	9	2	0.424
Ionosphere	351	34	2	0.399
Pima†	768	8	2	0.427
Sonar	208	60	2	0.095
Seeds	210	7	3	0.000
Protein	116	20	6	0.330
Optdigits-389	283	64	3	0.032
Letters-IJLT	250	16	4	0.168

†Data is normalized using the standard deviation.

$\{3, 8, 9\}$, generated by sampling each instance with a probability of 0.15. *Letters-IJLT* is a randomly sampled subset of 250 instances from the letter recognition data set containing only the letters $\{I, J, L, T\}$.

Table 1 reports the number of instances, features, and classes. The coefficient of variation (CV) [14] describes the degree of class imbalance: zero indicates perfectly balanced classes, while higher values indicate higher class imbalance.

5.2 Algorithms

For problems with pairwise constraints, we compare our model to MPCK-Means[4] and LCVQE.[5] For problems with partition-level constraints, we compare our model to PLCC.[6] For MPCK-Means and PLCC we used the weights proposed in the original papers. Increasing the weights did not lead to a significant change in results.

If K=2, we use the efficient QUBO encoding (Eq. (16)). Otherwise, we use the general QUBO model (Eq. (10)).

5.3 Evaluation Measures

Since labels are available for the data sets, we use the following measures to evaluate and compare the different methods.

Adjusted Rand Index (ARI) Rand Index [30] measures agreement between two partitions of the same data, P_1 and P_2. Each partition represents $\binom{n}{2}$ decisions over all pairs, assigning them to the same or different clusters. Let a be the number of pairs assigned to the same cluster in both P_1 and P_2. Let b be the number of pairs assigned to different clusters. Rand Index is defined as follows:

$$RI(P_1, P_2) = \frac{a + b}{\binom{n}{2}},$$

[4] Obtained from www.cs.utexas.edu/users/ml/risc/code.

[5] Obtained from github.com/danyaljj/constrained_clustering.

[6] As the code is not available, we implemented PLCC in Python.

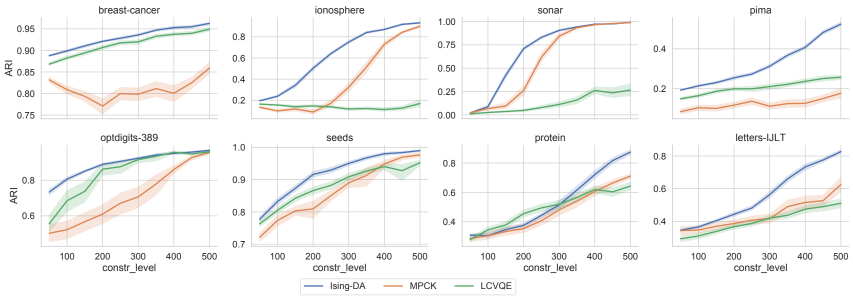

Fig. 1. Comparison of ARI scores for clustering with pairwise constraints.

while the Adjusted Rand Index (ARI) [17] is a correction for RI, based on its expected value:

$$ARI = \frac{RI - \mathbb{E}(RI)}{Max(RI) - \mathbb{E}(RI)}.$$

An ARI of zero indicates the partition is not better than a random assignment, while one indicates a perfect match.

Normalized Mutual Information (NMI) Mutual information quantifies the statistical information shared between two distributions [31]. We use $MI(P_1, P_2)$ to denote the mutual information between partitions P_1 and P_2, and $H(P_i)$ to denote the entropy of partition P_i. Normalized mutual information (NMI) [31] is normalized using a generalized mean (e.g., arithmetic or geometric) of $H(P_1)$ and $H(P_2)$:

$$NMI(P_1, P_2) = \frac{MI(P_1, P_2)}{Mean(H(P_1), H(P_2))}$$

Values close to zero indicate independent partitions, while values close to one indicate a significant agreement between $P1$ and $P2$. We use NMI based on arithmetic mean.

Fraction of violated constraints We compute the mean fraction of constraints that were violated in the partition.

5.4 Empirical Results

Instance-level Pairwise Constraints We compare our framework with MPCK-Means and LCVQE, on clustering with different numbers of randomly generated pairwise constraints. Following Covões et al. [9], each constraint is generated by randomly selecting two different instances in the data set and adding an ML constraint if they are in the same cluster and a CL constraint otherwise.

Figure 1 shows the performance for a varying number of pairwise constraints, measured by ARI. Each point in the plot is the average of 50 runs with different,

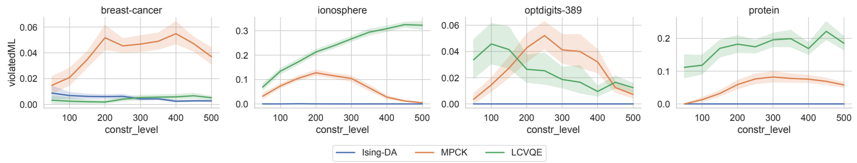

Fig. 2. Average fraction of violated must-link constraints.

randomly generated, sets of constraints. The bands represent the 95% confidence interval obtained using bootstrapping with 1000 replications. Note that the graphs *do not* share the same y-axis to increase readability (each graph presents data for a different data set and we do not compare across data sets). Results for NMI exhibited similar patterns and are omitted due to space.

In all cases but one, our framework outperforms the other methods. In *Breast Cancer, Ionosphere, Sonar, Pima, Optdigits-389, Seeds, Letters-IJLT* our framework is at least as good, and usually significantly better, across all numbers of constraints. In *Protein* there is no dominating algorithm and our framework is the best performing one for problems with large number of constraints (approximately 300 or more) while LCVQE is the best performing algorithm for problems with smaller number of constraints (less than 300).

Interestingly, there is no clear winner between LCVQE and MPCK-Means. In three data sets LCVQE outperforms MPCK-Means, in two data sets MPCK-Means outperforms LCVQE, and in the rest they are comparable. In contrast, our framework clearly outperforms the other methods.

Figure 2 shows the average fraction of violated must-link constraints and the 95% confidence interval for four data sets. In all data sets but *Breast Cancer*, we find that our method violates fewer constraints than the other methods, and in most cases does not violate any of the constraints. On *Breast Cancer*, our method and LCVQE outperform MPCK-Means, but do not dominate each other. Analysis of violated CL constraints is omitted due to space. As with ML constraints, our method is as good or better than the other methods in all cases except for *Breast Cancer*.

Partition-level Constraints We compare our framework with PLCC on clustering with different numbers of randomly generated partition-level constraints, taken from the true labels. To be consistent with previous work [23, 24], we present the number of constraints as the fraction of the labeled data points. Figure 3 shows the performance of PLCC and our algorithm, measured by ARI. Results for NMI exhibit similar patterns and are omitted due to space.

Our method is consistently at least as good as PLCC, and in most cases better. Interestingly, for PL constraints, the improvement observed for general clustering problems is larger than the one observed for problems with $K = 2$.

Next, we analyze the fraction of violated partition-level constraints. When $K = 2$, we found that both algorithms satisfy approximately 100% of the con-

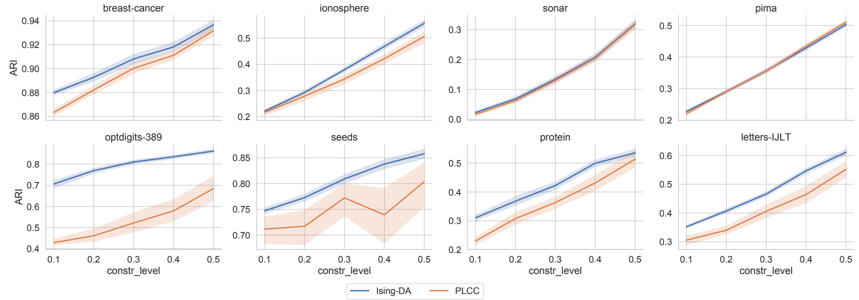

Fig. 3. ARI scores for clustering with partition-level constraints.

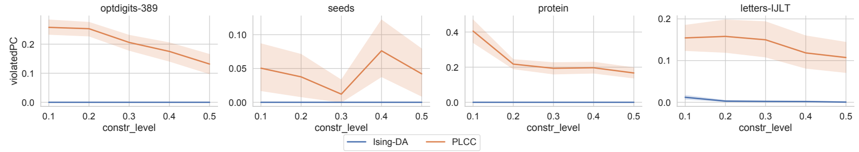

Fig. 4. Average fraction of violated partition-level constraints for $K>2$.

straints, with no significant differences. For the data sets with $K > 2$, PLCC violates a significant portion of the partition-level constraints while our method continues to satisfy all of them (see Figure 4). This may account for the larger difference in performance between the two algorithms for data sets with $K > 2$.

Mixed Constraint Types One of the advantages of our method, based on a mathematical model solved using a general optimization technique, is the ability to easily combine different types of constraints without the need to create a specialized algorithm.

To demonstrate this ability, we present results for problems that involve both pairwise and partition-level constraints. As far as we are aware, such problems cannot be solved by any existing heuristic techniques. Figure 5 reports

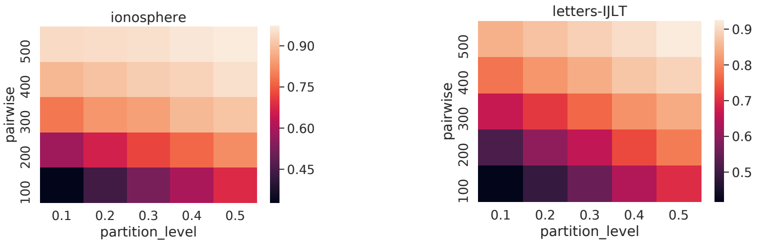

Fig. 5. ARI for mixed constraints.

the ARI for *Ionosphere* and *Letters-IJLT* for different combinations of pairwise and partition-level constraints. We can see that fusing different types of side-information can improve the clustering performance. Results for the other data sets exhibit similar patterns and are omitted due to space.

5.5 Comparison to Exact Methods

Despite the differences, it may be of interest to compare our approach to exact techniques. In this section, we compare our Ising framework to the CP approach with similar objective function [12] based on both the objective value and the accuracy of obtained solutions. We use the original code that is implemented in the Gecode solver[16] [7] and compare the solutions obtained by the DA after 5 seconds to the solutions found by Gecode with a time limit of 500 seconds. Note that objective value is only comparable in case both methods satisfy the same set of constraints. For CP, solutions that satisfy all pairwise constraints were found for all instances. For DA, solutions that satisfy all pairwise constraints were found for 595 out of the 600 instances. In each of the other five instances only a single pairwise constraint was not satisfied, however we remove these instances when comparing the objective values.

Table 2. Comparison between our Ising approach and Constraint Programming.

Data	Num.	`Ising-DA (5s)`		`CP (500s limit)`			DA/CP
set	constr.	Obj.	ARI	Obj.	ARI	% Opt	Obj.
Sonar (K=2)	50	31764.7	0.02	32992.2	0.04	0%	0.9632
	150	35501.9	0.43	35760.6	0.41	0%	0.9927
	350	36588.7	0.94	36588.5	0.94	100%	1.0000
Ionosphere (K=2)	50	433763.2	0.20	464745.6	0.13	0%	0.9344
	150	478007.9	0.34	500041.8	0.30	0%	0.9566
	350	514919.0	0.84	514922.3	0.84	94%	1.0000
Optdigits (K=3)	50	18790486.7	0.73	18840370.7	0.73	0%	0.9974
	150	18862921.9	0.85	18947962.6	0.87	0%	0.9955
	350	18955340.4	0.94	18957524.6	0.94	0%	0.9999
Protein ($K = 6$)	50	226791.6	0.31	260764.2	0.23	0%	0.8701
	150	245273.6	0.35	270906.9	0.29	0%	0.9070
	350	259862.0	0.62	269950.0	0.56	6%	0.9643

Table 2 shows the average objective value (lower is better) and ARI (higher is better) obtained by each of the approaches on four data sets with different characteristics and a varying number of pairwise constraints. We also list the percentage of instances for which CP was able to prove optimality and the average per-instance objective ratio between the two methods (DA/CP). In the

[7] Obtained from cp4clustering.github.io.

majority of cases Gecode was not able to prove optimality within the time limit. Furthermore, solutions found by the DA within 5 seconds are approximately equal or better for all configurations. In terms of clustering accuracy (measured by ARI), our approach outperforms CP for Ionosphere and Protein while in Sonar and Optdigits the methods are comparable.

5.6 Comparison to CPU baselines

Our interest in Ising models is motivated by their ability to be efficiently solved by a variety of specialized hardware platforms. To demonstrate the benefit of specialized hardware, we compare the results of the DA, a CMOS annealer, to two CPU baselines for Ising models: `neal`, a simulated annealer for Ising models, and `qbsolv`, a decomposing solver that splits QUBO problems into sub-problems solved by a tabu search (both are part of D-Wave's Ocean software package).[8]

We compare the quality of solutions obtained by these tools after 10 seconds and after 30 seconds to the solutions obtained by the DA after one and 5 seconds. Table 3 reports the mean ARI for four selected data sets for different numbers of pairwise constraints. Solutions that violate the one-hot encoding are considered to have an ARI of zero. As solutions obtained by the CPU solvers often do not satisfy all constraints, we do not compare the methods based on objective value.

Table 3. Mean ARI for DA vs. CPU solvers.

Num	`DA`		`neal`		`qbsolv`	
const.	1s	5s	10s	30s	10s	30s
50	**.02**	**.02**	**.02**	**.02**	**.02**	**.02**
150	.41	**.43**	.38	.40	.39	.40
350	**.94**	**.94**	**.94**	**.94**	**.94**	**.94**

(a) Sonar (K=2)

Num	`DA`		`neal`		`qbsolv`	
const.	1s	5s	10s	30s	10s	30s
50	.19	**.20**	.18	.19	.16	.16
150	.33	**.34**	.26	.31	.32	.33
350	**.84**	**.84**	.82	.83	.80	.82

(b) Ionosphere (K=2)

Num	`DA`		`neal`		`qbsolv`	
const.	1s	5s	10s	30s	10s	30s
50	.71	**.73**	.35	.48	.27	.29
150	**.85**	**.85**	.57	.72	.32	0.35
350	**.94**	**.94**	.89	.91	.82	0.83

(c) Optdigits (K=3)

Num	`DA`		`neal`		`qbsolv`	
const.	1s	5s	10s	30s	10s	30s
50	.29	**.31**	.02	.02	.26	.27
150	.30	**.35**	.06	.05	.28	.31
350	.59	.62	.32	.37	.61	**.63**

(d) Protein (K=6)

We can see that the DA achieves better performance compared to the CPU baseline, even when we allow the CPU baselines longer time limits. In all but one configuration, DA with 5 seconds outperforms `neal` and `qbsolv` with 30 seconds. Interestingly, even when given only one second the DA performs well

[8] Both tools obtained from github.com/dwavesystems.

and in most configurations obtain solutions that are equal or better than those found by the CPU baselines in 30 seconds.

6 Discussion & Limitations

Our empirical evaluation shows that our method, based on an Ising model and specialized hardware, outperforms state-of-the-art K-Means-like methods. In unsupervised clustering, Kumar et al. [22] found that using Ising models for clustering achieves, at best, equal performance to K-means. Our results suggest that in the semi-supervised setting, where the problems include a set of constraints, using specialized hardware is a promising direction. The comparison to CP and CPU baselines shows that our approach can provide high quality solutions fast, making it an attractive solution for modern data mining tasks.

Our framework can be extended to other scenarios: representing new types of constraints (e.g., cluster-size constraints [7]), tuning the weights of the constraints if they are not fully-trusted, and evaluating our model with constraints arising from active learning [3] are all potential extensions of our work. While our models can incorporate any constraint that can be represented as a quadratic equality or inequality over the binary variables, some constraints may require additional auxiliary or slack variables. Investigating ways to efficiently encode other types of constraints is also an interesting direction for future work.

Our method is sensitive to hardware-related limitations. For example, the number of data points is limited by the number of variables supported by the hardware and our ability to represent the objective is limited by the precision. However, new hardware allows for larger problems and increased precision (e.g., [37, 1]) and improved optimization schemes can reduce the need to tune the temperature schedule and potentially yield superior performance [20].

Our model can be solved on any platform that supports Ising models. As a large number of novel computational platforms (including quantum computers) have chosen Ising as their main abstraction [8], experimenting with new and different hardware platforms is an important direction of future work.

7 Conclusion

We address the problem of semi-supervised clustering on specialized hardware and present an Ising formulation that can be solved on a variety of novel hardware platforms. Our empirical analysis shows that our method outperforms the state-of-the-art heuristic methods for semi-supervised clustering and, unlike those algorithms, can support combinations of constraint types. The use of a mathematical model means that our framework is easily extended to support other types of constraints and hardware platforms.

Acknowledgements This research was supported by the Natural Sciences and Engineering Research Council of Canada, Fujitsu Co-Creation Research Laboratory at the University of Toronto, and the Lyon Sachs Fellowship.

References

1. Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., Katzgraber, H.G.: Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. Frontiers in Physics **7** (2019)
2. Babaki, B., Guns, T., Nijssen, S.: Constrained clustering using column generation. In: CPAIOR. pp. 438–454 (2014)
3. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: SDM. pp. 333–344. SIAM (2004)
4. Basu, S., Davidson, I., Wagstaff, K.: Constrained clustering: Advances in algorithms, theory, and applications. CRC Press (2008)
5. Bian, Z., Chudak, F., Macready, W.G., Rose, G.: The ising model: teaching an old problem new tricks. D-wave systems **2** (2010)
6. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML. pp. 81–88 (2004)
7. Bradley, P., Bennett, K., Demiriz, A.: Constrained k-means clustering. Microsoft Research, Redmond **20**(0), 0 (2000)
8. Coffrin, C., Nagarajan, H., Bent, R.: Evaluating ising processing units with integer programming. In: CPAIOR. pp. 163–181 (2019)
9. Covões, T.F., Hruschka, E.R., Ghosh, J.: A study of k-means-based algorithms for constrained clustering. Intelligent Data Analysis **17**(3), 485–505 (2013)
10. Dao, T.B.H., Duong, K.C., Vrain, C.: A declarative framework for constrained clustering. In: ECML-PKDD. pp. 419–434 (2013)
11. Dao, T.B.H., Duong, K.C., Vrain, C.: Constrained minimum sum of squares clustering by constraint programming. In: CP. pp. 557–573 (2015)
12. Dao, T.B.H., Duong, K.C., Vrain, C.: Constrained clustering by constraint programming. Artificial Intelligence **244**, 70–94 (2017)
13. Davidson, I., Ravi, S.: Clustering with constraints: Feasibility issues and the k-means algorithm. In: SDM. pp. 138–149 (2005)
14. DeGroot, M.H., Schervish, M.J.: Probability and Statistics. Pearson Education (2012)
15. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
16. Gecode Team: http://www.gecode.org
17. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification **2**(1), 193–218 (1985)
18. James, G., Witten, D., Hastie, T., Tibshirani, R.: An introduction to statistical learning. Springer (2013)
19. Johnson, M.W., Amin, M.H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A.J., Johansson, J., Bunyk, P., et al.: Quantum annealing with manufactured spins. Nature **473**(7346), 194 (2011)
20. Katzgraber, H.G., Trebst, S., Huse, D.A., Troyer, M.: Feedback-optimized parallel tempering monte carlo. Journal of Statistical Mechanics: Theory and Experiment **2006**(03), P03018 (2006)
21. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
22. Kumar, V., Bass, G., Tomlin, C., Dulny, J.: Quantum annealing for combinatorial clustering. Quantum Information Processing **17**(2), 39 (2018)
23. Liu, H., Fu, Y.: Clustering with partition level side information. In: IEEE ICDM. pp. 877–882 (2015)

24. Liu, H., Tao, Z., Fu, Y.: Partition level constrained clustering. IEEE TPAMI **40**(10), 2469–2483 (2018)
25. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)
26. Lucas, A.: Ising formulations of many np problems. Frontiers in Physics **2**, 5 (2014)
27. Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The planar k-means problem is np-hard. In: International Workshop on Algorithms and Computation. pp. 274–285 (2009)
28. Matsubara, S., Tamura, H., Takatsu, M., Yoo, D., Vatankhahghadim, B., Yamasaki, H., Miyazawa, T., Tsukamoto, S., Watanabe, Y., Takemoto, K., et al.: Ising-model optimizer with parallel-trial bit-sieve engine. In: CISIS. pp. 432–438 (2017)
29. Pelleg, D., Baras, D.: K-means with large and noisy constraint sets. In: ECML. pp. 674–682 (2007)
30. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association **66**(336), 846–850 (1971)
31. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research **3**, 583–617 (2002)
32. Trevor, H., Robert, T., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction. Springer (2009)
33. Tsukamoto, S., Takatsu, M., Matsubara, S., Tamura, H.: An accelerator architecture for combinatorial optimization problems. Fujitsu Sci. Tech. J **53**(5), 8–13 (2017)
34. Ushijima-Mwesigwa, H., Negre, C.F., Mniszewski, S.M.: Graph partitioning using quantum annealing on the d-wave system. In: Proceedings of the Second International Workshop on Post Moores Era Supercomputing. pp. 22–29. ACM (2017)
35. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: ICML. pp. 577–584 (2001)
36. Xing, E.P., Jordan, M.I., Russell, S.J., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: NIPS. pp. 521–528 (2003)
37. Yamaoka, M., Yoshimura, C., Hayashi, M., Okuyama, T., Aoki, H., Mizuno, H.: A 20k-spin ising chip to solve combinatorial optimization problems with cmos annealing. IEEE Journal of Solid-State Circuits **51**, 303–309 (2016)